



## Chương 2

### Mô hình liên kết thực thể

(Entity Relationship model – ER model)

1



### Mô hình liên kết thực thể

- Mô hình ER - Các khái niệm
- Tầm quan trọng của mô hình hóa dữ liệu
- Lược đồ liên kết thực thể
  - Thực thể (Entity)
  - Thuộc tính – Attribute

2



## Mô hình ER - Các khái niệm

- Mô hình ER được dùng để xây dựng mô hình hóa dữ liệu ý niệm (conceptual data modeling) nhằm biểu diễn cấu trúc và các ràng buộc của CSDL.
- Mô hình ER như một công cụ giao tiếp giữa người thiết kế CSDL và người sử dụng cuối cùng để xây dựng CSDL trong giai đoạn phân tích. Nó độc lập với DBMS và quá trình thi công CSDL.

3



## Tầm quan trọng của mô hình hóa dữ liệu

- Giúp thu tóm các đặc tính dữ liệu, nắm bắt quy tắc nghiệp vụ và các ràng buộc trong quá trình mô hình hóa dữ liệu
- Thu thập dữ liệu
- Hệ thống được thiết kế theo hướng dữ liệu (data-oriented) sẽ ổn định hơn hướng xử lý dữ liệu (process oriented)

4



## Lược đồ liên kết thực thể

- Mô hình ER được diễn tả bằng lược đồ liên kết thực thể (Entity Relationship Diagram – ERD)
- Ba phần tử cơ bản:
  - Thực thể (Entity)
  - các thuộc tính (Attributes).
  - Mối liên kết (Relationship)

5



## Thực thể (Entity)

- Thực thể là đối tượng chính mà ta thu thập thông tin xoay quanh chúng.
- Thực thể có thể là
  - Một người như nhân viên, sinh viên,...
  - Một nơi chốn như thành phố, đất nước,...
  - Một sự kiện như mua hàng, trả lương,...
  - Một khái niệm như môn học, tài khoản,...

6



## Kiểu thực thể (Entity Type)

- Kiểu thực thể: là một tập hợp các thực thể có cùng tính chất. Kiểu thực thể được biểu diễn bằng hình chữ nhật.
- Thể hiện thực thể (entity instance): là một trường hợp cụ thể của kiểu thực thể đó.
  - Ví dụ: kiểu thực thể Customer có các điển hình là Tom và Peter. Mỗi Customer đều có mã khách khác nhau, và có thể thực hiện các dịch vụ như đặt hàng, thanh toán tiền ....

7



## Kiểu thực thể (Entity Type)

- Các kiểu thực thể:
  - Thực thể mạnh (strong entity type): tồn tại độc lập với những kiểu thực thể khác
  - Thực thể yếu (weak entity type): tồn tại phụ thuộc vào kiểu thực thể khác
  - Ví dụ:
    - EMPLOYEE là kiểu thực thể mạnh.
    - DEPENDENT là kiểu thực thể yếu, lệ thuộc vào EMPLOYEE.

8



## Cách đặt tên và ký hiệu thực thể

- Mỗi kiểu thực thể phải có một tên gọi, nên là danh từ số ít và viết chữ hoa.
- Ký hiệu của các kiểu thực thể

EMPLOYEE

Thực thể mạnh

DEPENDENT

Thực thể yếu

9



## Thuộc tính - Attribute

- Thuộc tính là Oặc tính của một thực thể hay một mối liên kết.
  - Ví dụ: kiểu thực thể STUDENT có các thuộc tính như Student\_ID, Student\_Name, Address, Major.  
*STUDENT Student\_ID, Student\_Name,...*
  - Thuộc tính được biểu diễn bằng *hình bầu dục (oval)*.

10



## Các loại thuộc tính

- **Thuộc tính đơn** (simple attribute):
  - là thuộc tính không thể phân nhỏ được.
  - Ví dụ: Color, Weight, HorsePower
- **Thuộc tính phức hợp** (composite attribute):
  - là thuộc tính có thể phân thành nhiều thành phần.
  - Ví dụ: Thuộc tính Address bao gồm các thành phần Street, District, City
- **Thuộc tính đơn trị** (single valued attribute)
- **Thuộc tính đa trị** (multivalued attribute):
  - có thể có nhiều hơn một trị cho một thể hiện của thực thể. Biểu diễn bằng *hình bầu dục nét đứt*.
  - Ví dụ: Thực thể COURSE có thuộc tính Teacher đa trị, một môn học có thể được dạy bởi nhiều hơn 1 thầy cô.

11



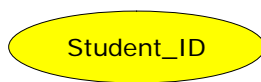
## Các loại thuộc tính

- **Thuộc tính xác định/khóa** ( identifier/key attribute):
  - là 1 thuộc tính hoặc 1 tổ hợp các thuộc tính xác định được các thể hiện riêng biệt của 1 kiểu thực thể
  - Ví dụ: Student\_ID là thuộc tính xác định của kiểu thực thể STUDENT
- **Thuộc tính dẫn xuất** (derived attribute):
  - là thuộc tính mà trị của nó có thể tính ra được từ các thuộc tính khác. Biểu diễn bằng *hình bầu dục nét đứt*.
  - Ví dụ: Year\_Employed là thuộc tính dẫn xuất từ thuộc tính Date\_Employed
- **Thuộc tính chứa** (stored attribute)

12

## Cách đặt tên và ký hiệu

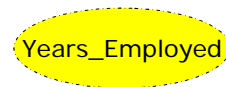
- Mỗi thuộc tính nên được biểu diễn là danh từ số ít và viết chữ thường.
- Ký hiệu của các kiểu thuộc tính:



Thuộc tính xác định



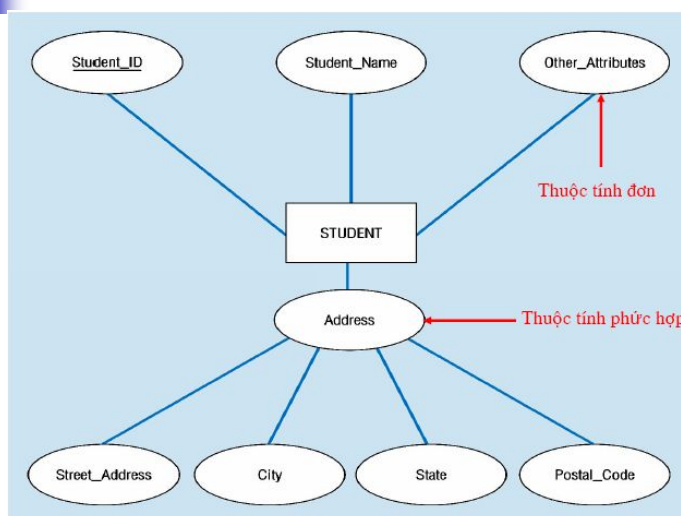
Thuộc tính đa trị



Thuộc tính dẫn xuất

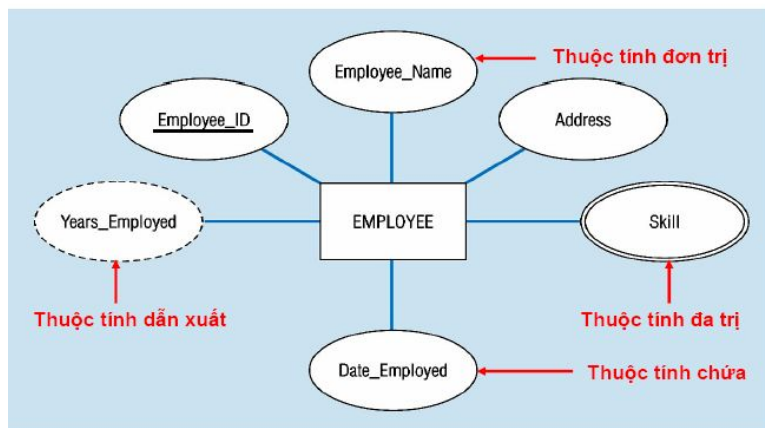
13

## ERD – Ví dụ về thuộc tính



14

## ERD- Ví dụ về thuộc tính



15

## Khóa (Key)-Các khái niệm liên quan

- Khóa / thuộc tính xác định (key / identifier)
  - Khái niệm
- Thuộc tính khóa (key attribute / prime attribute / identifier attribute) và thuộc tính không khóa (non-key attribute)
- Thuộc tính không khóa còn được gọi là thuộc tính mô tả (descriptor attribute).

16

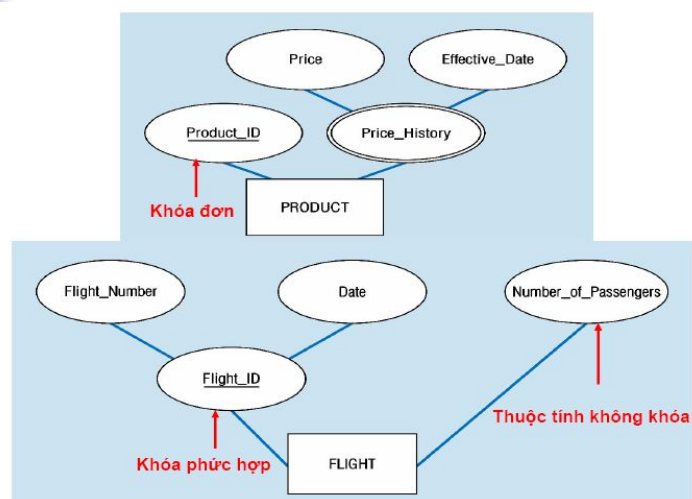


## Khóa (Key)-Các khái niệm liên quan

- Khóa đơn (simple key).
- Khóa phức hợp (composite key)
- Khóa dự tuyển (candidate key)
  - Một kiểu thực thể có ít nhất một khóa dự tuyển.
- Khóa chính (primary key)
  - Khóa chính là một khóa tiêu biểu trong các khóa dự tuyển của một kiểu thực thể.
  - Một kiểu thực thể chỉ có một khóa chính.
  - Khóa chính dùng để liên kết giữa các thực thể

17

## Khóa (Key)- Ví dụ



18



## Mối liên kết (Relationship)

- **Mối liên kết (relationship)** diễn tả sự kết hợp giữa một hay nhiều kiểu thực thể với nhau.
- **Kiểu liên kết (relationship type)** là một sự kết hợp có ý nghĩa giữa các kiểu thực thể. Được biểu diễn bằng hình thoi
- **Một thể hiện (instance)** của một kiểu liên kết là một sự kết hợp giữa các thể hiện của các kiểu thực thể tham gia vào mối liên kết đó

19

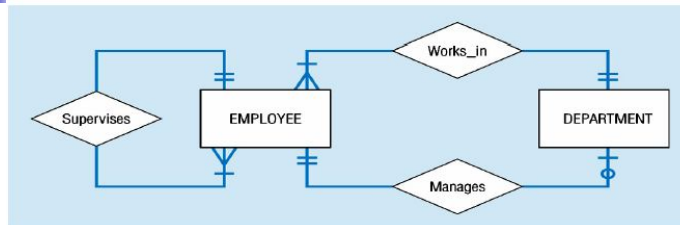


## Mối liên kết (Relationship)

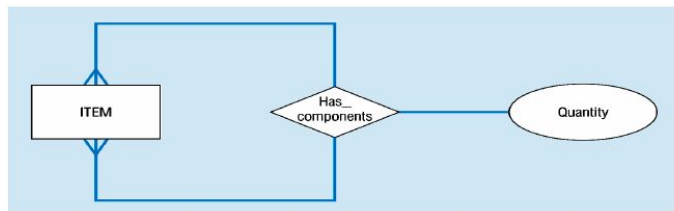
- Mỗi liên kết có thể có nhiều thuộc tính dùng để mô tả các đặc tính của sự liên kết giữa các thực thể;
- Giữa các thực thể có thể có *nhiều* kiểu mối liên kết

20

## Mối liên kết (Relationship)-VD



Hai thực thể có nhiều mối liên kết



21

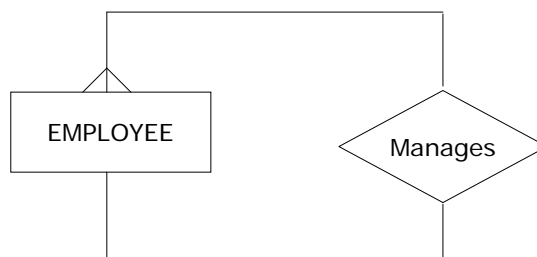
## Mối liên kết (Relationship)

- Bậc/ngôi của mối liên kết (*degree/arity of relationship*): số lượng kiểu thực thể tham gia *đồng thời* vào mối liên kết này.
- Các loại mối liên kết
  - Mối liên kết 1-ngôi (*unary relationship*):
  - Mối liên kết 2-ngôi (*binary relationship*)
  - Mối liên kết 3-ngôi (*ternary relationship*): 3 kiểu thực thể *đồng thời* tham gia vào mối liên kết.

22

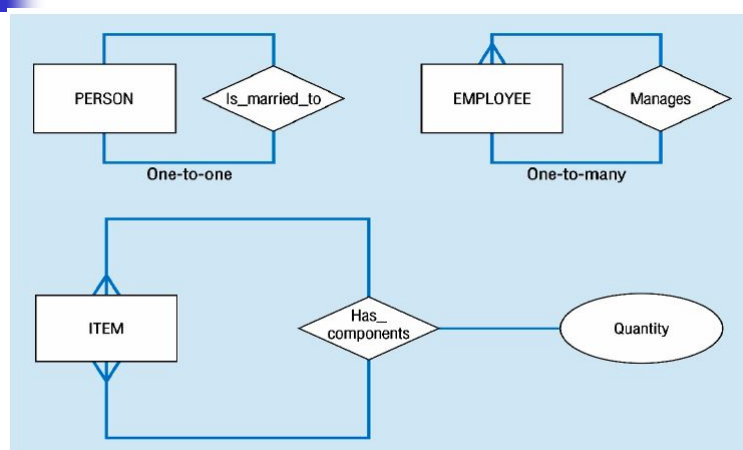
## Liên kết một ngôi(Unary relationship)

- Là mối liên kết giữa các thể hiện của cùng một kiểu thực thể; Còn gọi là mối liên kết đệ quy (recursive relationship)
- Ví dụ:



23

## Mối liên kết (Relationship)



Mối liên kết 1-ngôi

24

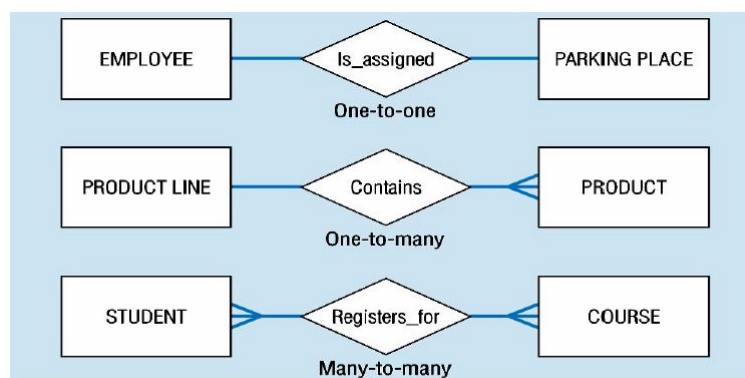
## Liên kết hai ngôi (Binary relationship)

- Là mối liên kết giữa hai kiểu thực thể



25

## Mối liên kết (Relationship)

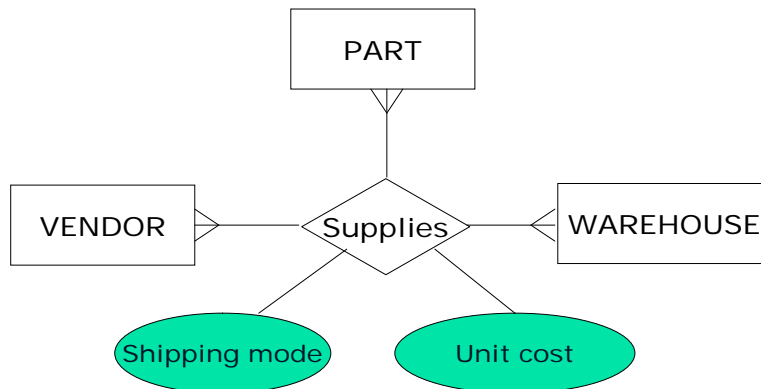


Mối liên kết 2-ngôi

26

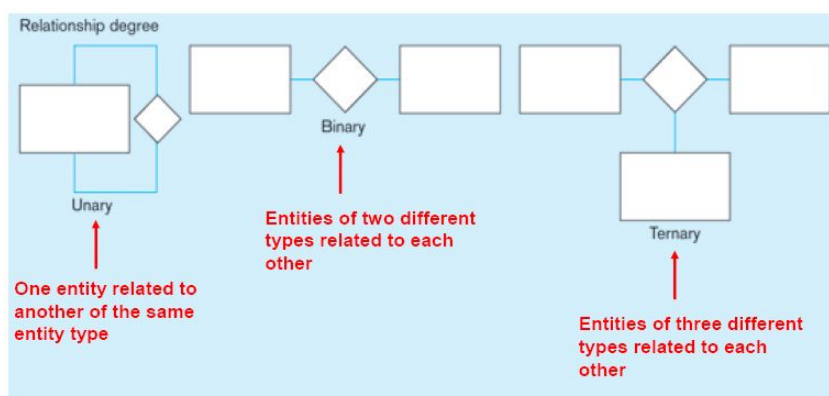
## Liên kết ba ngôi (Ternary relationship)

- Là mối liên kết giữa 3 kiểu thực thể




27

## Mối liên kết (Relationship)



28



## Lượng số của mỗi liên kết (*cardinality of relationship*)

- Lượng số của mỗi liên kết là *số lượng thể hiện thực thể* tham gia vào mỗi liên kết này.
- Các loại lượng số
  - một - một (one-to-one)
  - một - nhiều (one-to-many)
  - nhiều - nhiều (many-to-many)

29

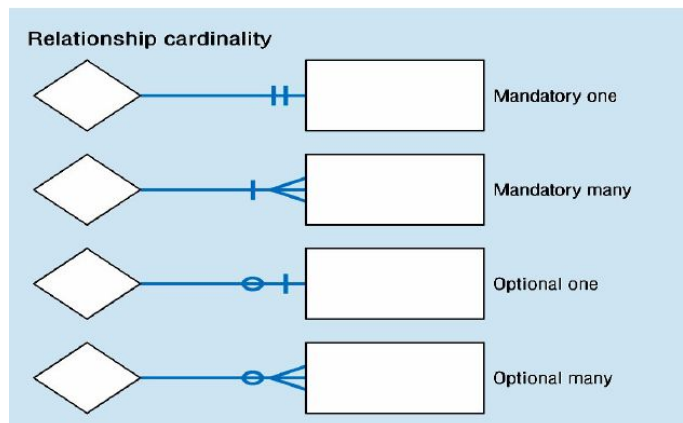


## Ràng buộc lượng số (*cardinality constraint*)

- Ràng buộc lượng số là số lượng thể hiện của thực thể này có thể hoặc phải liên kết với một thể hiện của thực thể khác.
- Lượng số nhỏ nhất
  - Nếu 0 là tùy chọn (optional).
  - Nếu một hoặc nhiều là bắt buộc (mandatory).
- Lượng số lớn nhất: số lượng thể hiện lớn nhất.

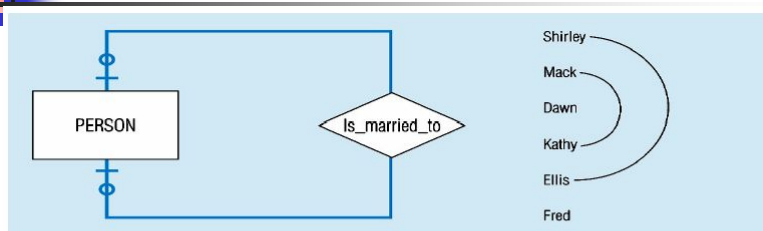
30

## Lượng số của mối liên kết (cardinality of relationship)

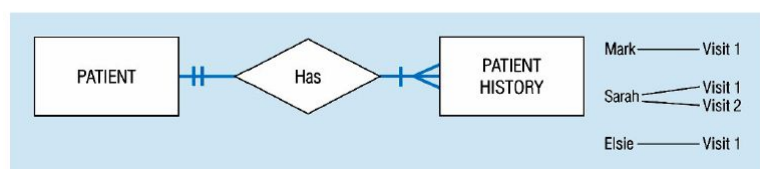


31

## Lượng số của mối liên kết (cardinality of relationship) -VD



Mối liên kết 1-ngôi một-một có lượng số tùy chọn

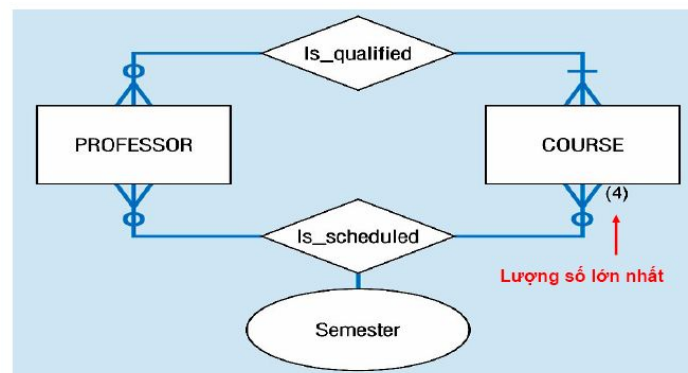


Mối liên kết 2-ngôi một-nhiều có lượng số bắt buộc

32



## Lượng số của mỗi liên kết (cardinality of relationship) -VD



Mỗi liên kết có lượng số tối đa xác định