

Chapter 1 : Khái niệm và kiến trúc của HCSDL

1.1: Đặc tính của (hàng tiếp cận) database:

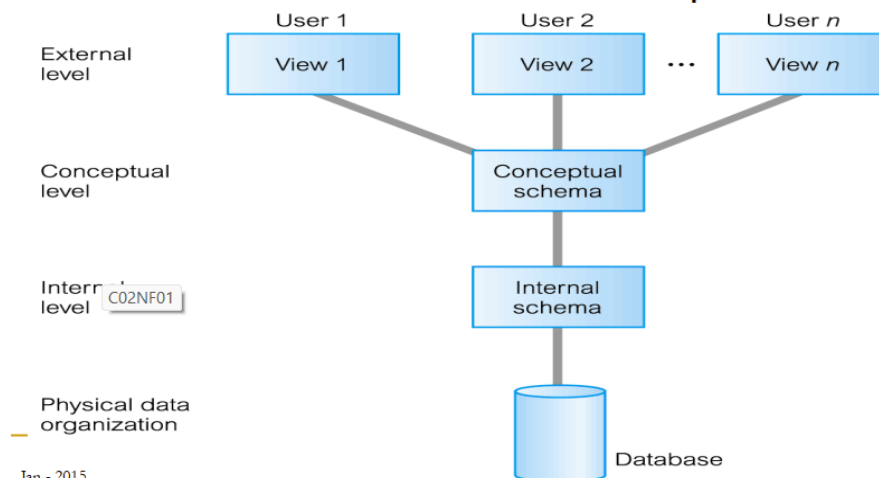
- Tạo miêu tả bản chất của hệ cơ sở dữ liệu.
- Tách biệt giữa chương trình và dữ liệu, trừu tượng hóa dữ liệu (data abstraction).
- Hệ tạo nhiều góc nhìn về dữ liệu (i.g, cùng một hệ csdl trường học, có nhiều góc nhìn khác về dữ liệu từ giáo viên và học sinh).
- Chia sẻ dữ liệu và thực thi truy vấn đồng bộ.

1.2: Kiến trúc ba lớp để và đặc lập dữ liệu

1.2.1: Objectives của kiến trúc ba lớp để:

- Mỗi user nên tiếp cận cùng một dữ liệu.
- User không cần biết về chi tiết lưu trữ csdl vật lý.
- DBA nên có khả năng thay đổi cấu trúc csdl mà không làm ảnh hưởng đến góc nhìn user.
- Cấu trúc bên trong của csdl phải không bị ảnh hưởng bởi những thay đổi về khía cạnh vật lý của kho lưu trữ.
- DBA nên có khả năng thay đổi cấu trúc quan niệm mà không ảnh hưởng tới mỗi user.

▪ Three-level architecture and data independence



1.2.2: Đặc lập dữ liệu:

Là khả năng có thể thay đổi lược đồ của một cấp (level) mà không cần thay đổi lược đồ cấp cao hơn.

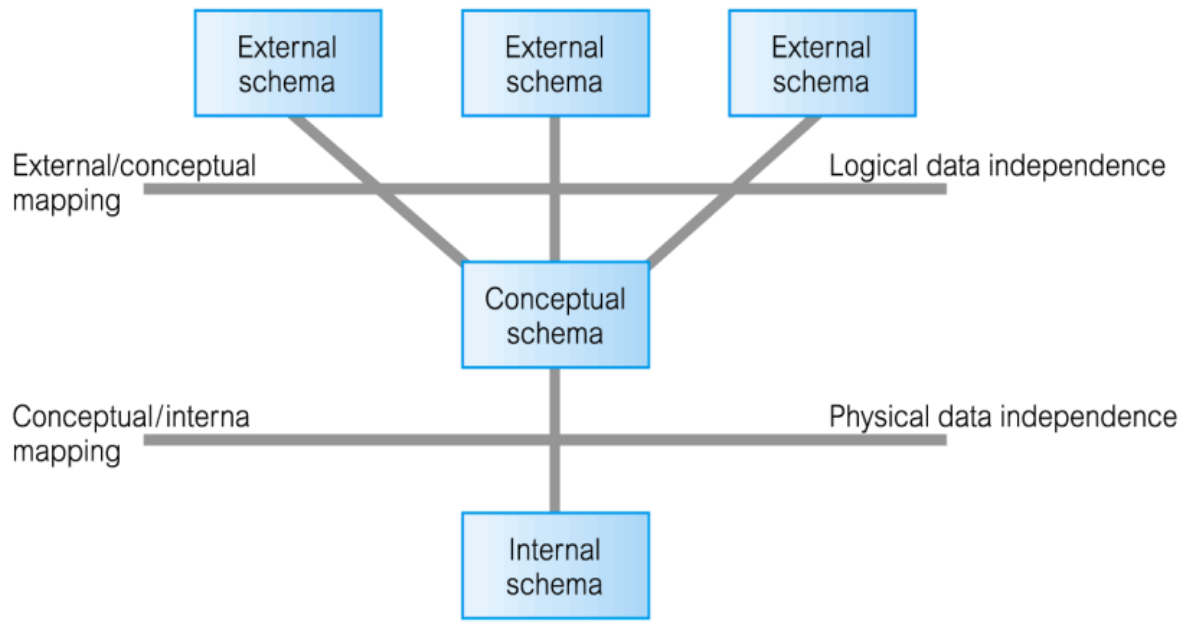
*Độc lập logic dữ liệu (logical data independence):

-Độc lập đến khả năng bất biến của các lược đồ ngoài đối với sự thay đổi của lược đồ khái niệm.

-Thay đổi về lược đồ quan niệm (thêm, bớt entity) không nên yêu cầu thay đổi đối với lược đồ ngoài, hoặc viết lại chương trình ứng dụng DB.

*Độc lập phần cứng (physical data independence):

-Độc lập đến khả năng bất biến của lược đồ khái niệm đối với sự thay đổi của lược đồ nội bộ.



1.3: Ngôn ngữ lập trình DB:

-Data Definition Language (DDL): hỗ trợ cho DBA hoặc user miêu tả các entity, attribute, quan hệ cần có cho ứng dụng cùng với những ràng buộc toàn vẹn và bảo mật.

-Data Manipulation Language (DML): cung cấp những thao tác xử lý dữ liệu trên dữ liệu trong DB.

-Data Control Language (DCL): định nghĩa những thao tác ngoài DDL, DML, i.g, cấp đặc quyền cho người dùng và xác định khi nào các thao tác được thực hiện không thể bị hủy ngang.

1.4: Mô hình, logic dữ liệu và trạng thái cơ sở dữ liệu (database state):

-Mô hình dữ liệu: một tập hợp toàn vẹn những khái niệm để miêu tả dữ liệu, quan hệ giữa các dữ liệu và điều kiện ràng buộc dữ liệu trong một tổ chức.

-Những loại mô hình dữ liệu:

+Cơ sở trên đối tượng: ERD, OO

+Cơ sở trên bản ghi: Relational, Network, Hierarchical

☞ Miêu tả dữ liệu ở mức khái niệm và ngoại vi.

+Phần cứng: dùng để mô tả dữ liệu ở cấp nội bộ (ctdl).

-Logic dữ liệu DB: là những miêu tả về DB, được thực hiện trong quá trình thiết kế và không nên thay đổi thông xuyên.

-Sơ đồ DB: là trình bày của logic dữ liệu DB trên giấy

-DB state (snapshot): dữ liệu trong DB trong một khoảng thời gian (/hoặc thời điểm) xác định.

Chương 2: Mô hình hóa dữ liệu sử dụng mô hình ER (Entity Relationship)

2.1: DB design process:

-Bao gồm hai hoạt động:

+Thiết kế DB: thiết kế lược đồ khái niệm cho ứng dụng DB

+Thiết kế chương trình ứng dụng: phần thuộc về công nghệ phần mềm.

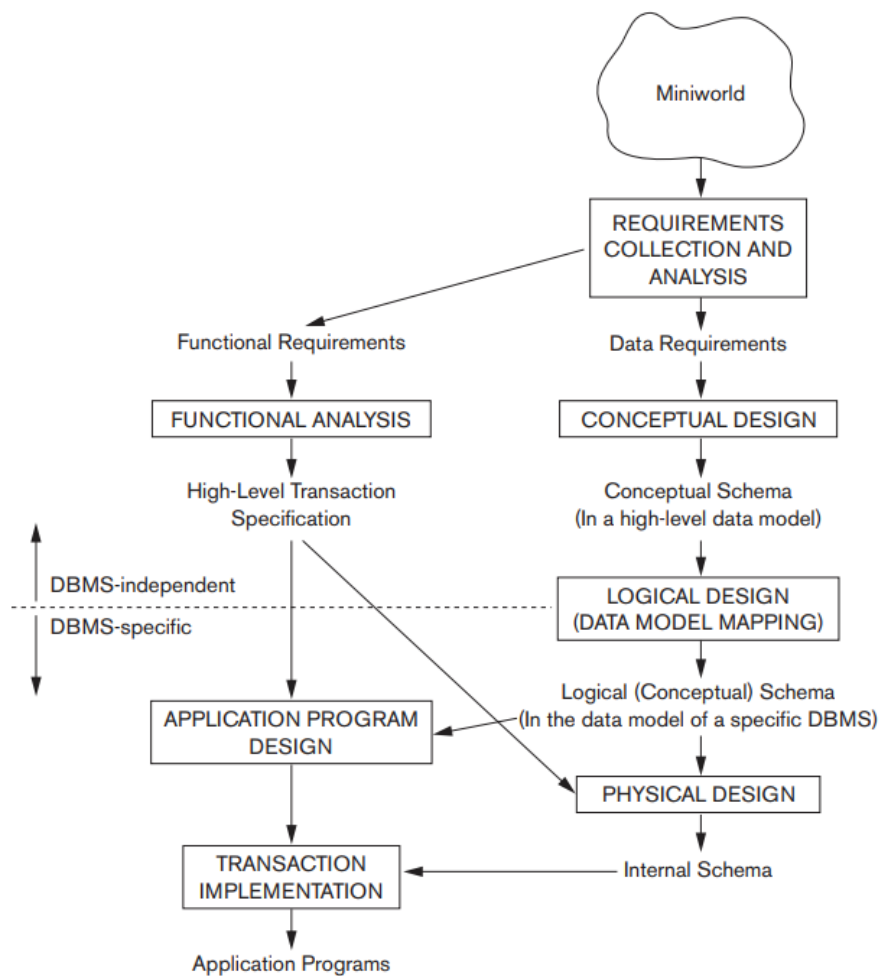


Figure 3.1
A simplified diagram to illustrate the main phases of database design.

2.2: Mô hình quan hệ thực thể (ER Model):

**Mô hình ER:*

-Mô hình khái niệm dữ liệu bậc cao phổ biến.

-Một tổ chức luận lý của dữ liệu trong hcsdl.

**Sơ đồ ER:*

Ký hiệu sơ đồ hóa tương ứng với mô hình ER.

2.3: Ví dụ về ứng dụng DB:

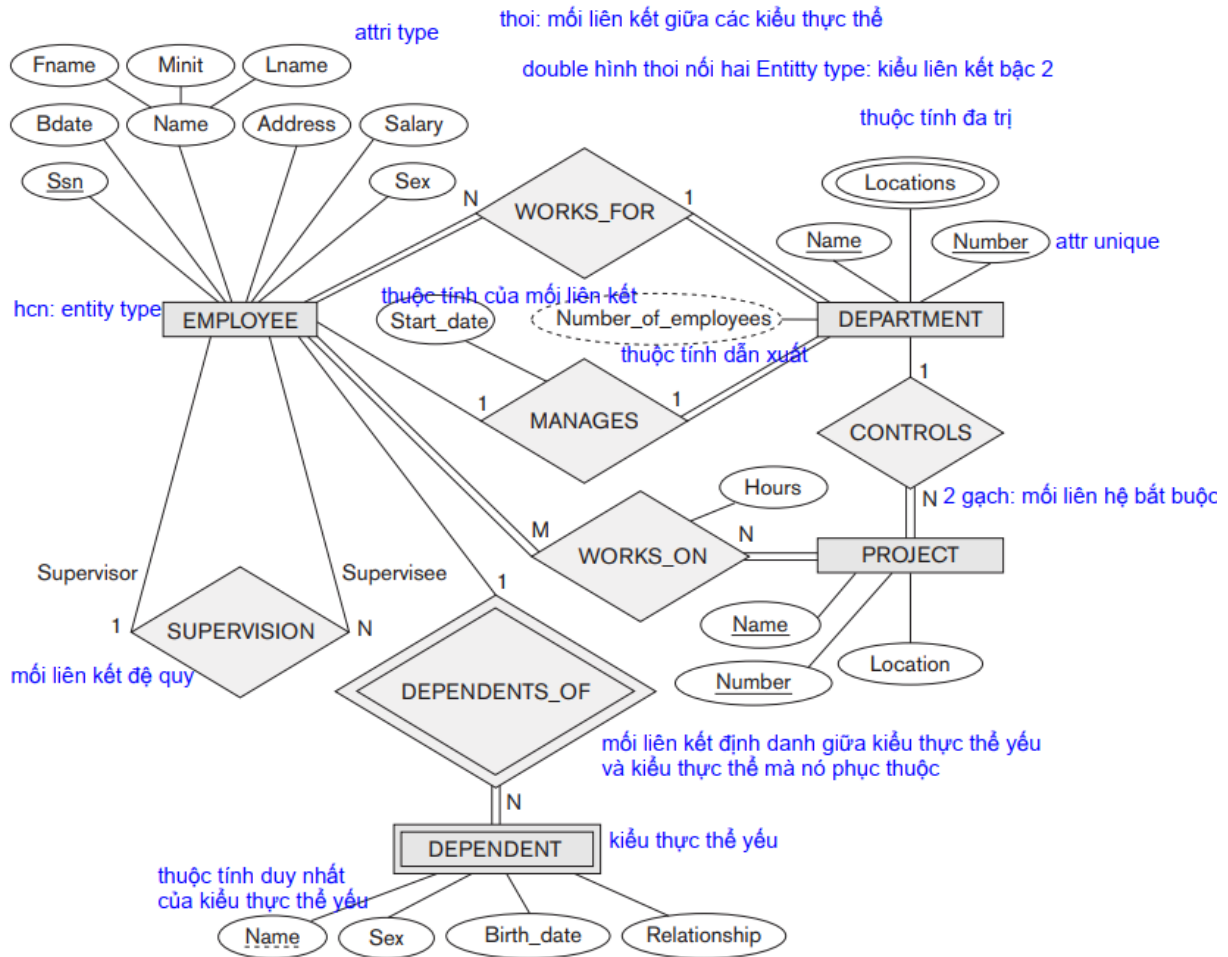
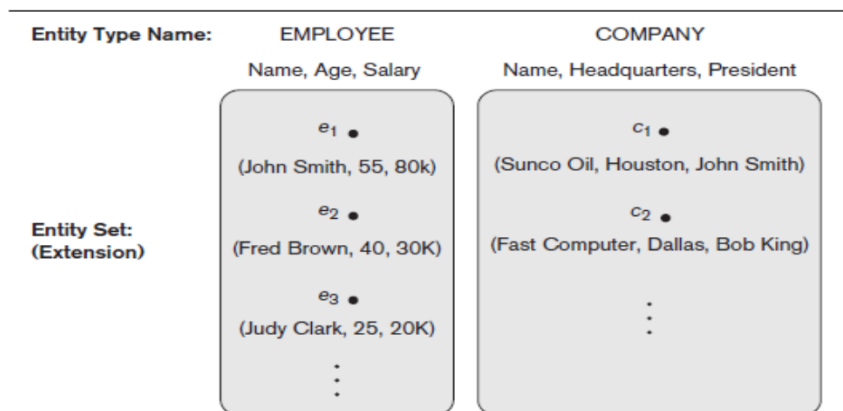


Figure 3.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 3.14.

2.4: Khái niệm mô hình ER:

-Loại Entity: tập các entity (thực thể) có cùng attribute.



-Ràng buộc khóa hoặc ràng buộc duy nhất (key constraint):

+Attribute của entity sao cho có giá trị riêng biệt (khác nhau) giữa các entity trong tập các entity (*entity type*).

-Một entity type có thể có nhiều hơn một khóa.

2.5: Các quan hệ và kiểu quan hệ (Relation type):

-Kiểu quan hệ R giữa n entity kiểu E_1, E_2, \dots, E_n .

+Biểu diễn tập các liên kết giữa các entity từ những kiểu entity này.

-Bậc của một kiểu quan hệ:

+Số kiểu thực thể tham gia.

+Binary (bậc 2), ternary (bậc 3) và n-array (bậc n).

-Có thể có nhiều kiểu quan hệ cho một tập các kiểu entity tham gia.

- Ex: MANAGES and WORKS_FOR are distinct relationships between EMPLOYEE and DEPARTMENT, but with different meanings and different relationship instances

2.6: Điều kiện ràng buộc trên kiểu quan hệ bậc 2:

-Tập số lượng: xác định số lượng tối đa của relationship instance mà entity có thể tham gia trong một quan hệ bậc hai.

+one-to-one (1:1)

+one-to-many (1:N) hoặc many-to-one (N:1)

+many-to-many (M:N)

2.7: Ràng buộc trên quan hệ bậc 2:

Lập thành viên (ràng buộc tham gia): xác định số lần tối đa của entity phụ thuộc vào việc nó có quan hệ với entity khác.

Summary of the Notation for ER Diagrams

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

Figure 7.14
Summary of the notation for ER diagrams.

Jan - 2015

CHƯƠNG 3: EERD

3.1 Subclasses, superclasses & Inheritance:

Mối quan hệ class/ sub class thường được gọi là quan hệ IS-A (IS-AN).

-Mỗi thành viên của subclass cũng là một entity trong superclass, nhưng với một vai trò cụ thể riêng biệt.

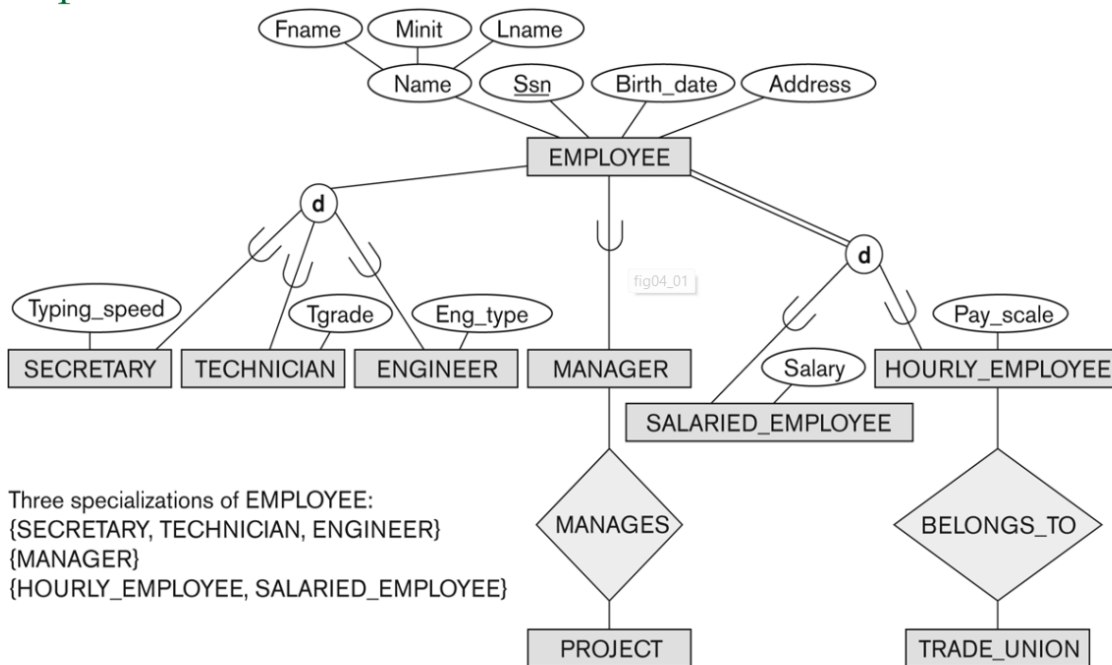
-Mỗi entity không tồn tại trong db không chỉ với tư cách là thành viên của lớp con, nó cũng phải là thành viên của superclass.

-Mỗi entity thuộc superclass có khả năng là thành viên của subclass.

-Không nhất thiết mỗi entity của superclass đều phải thuộc một subclass nào đó.

-Superclass và subclass là quan hệ one-to-one (1:1).

EER diagram notation to represent subclasses & specialization



Ian - 2015

9

-Subclass entity không thừa nhận tính (attribute) và quan hệ của superclass.

-Lưu ý rằng subclass với các attribute và mối quan hệ của nó (hoặc các bộ) của riêng nó cùng với tất cả các thuộc tính và mối quan hệ mà nó không thừa nhận từ superclass, có thể được gọi là một loại thực thể theo đúng nghĩa.

3.2: Chuyên hóa và tổng quan hóa:

3.2.1: Chuyên hóa – Specialization:

-Quá trình định nghĩa một tập con của một entity type, gọi superclass.

-Định nghĩa trên cơ sở là vài thuộc tính riêng biệt của các entity trong superclass.

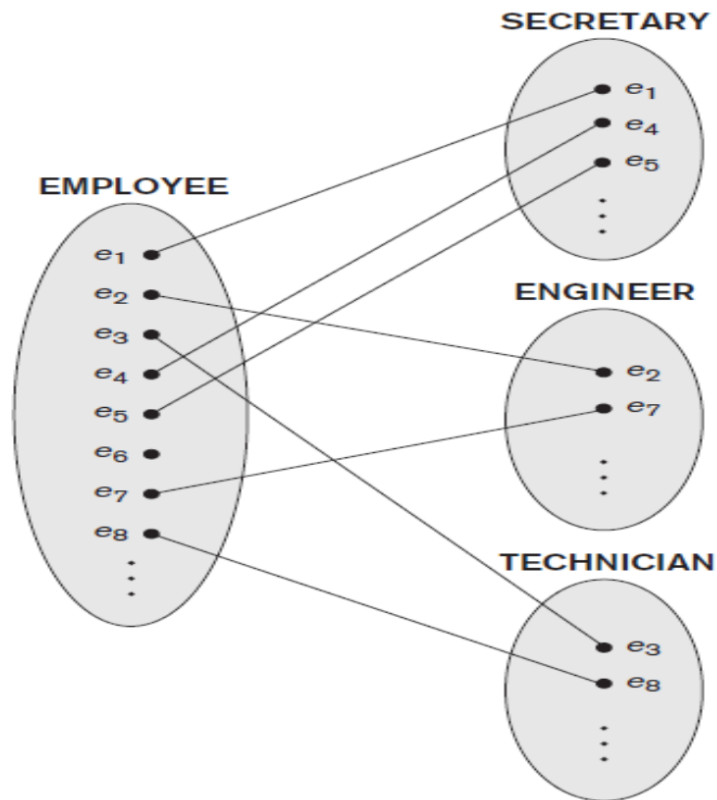
-Có thể có nhiều quá trình chuyên hóa của cùng một entity type, dựa trên những đặc trưng khác nhau.

*Subclass sở hữu:

-Thuộc tính của nó (các bộ).

-Kiểu quan hệ của nó.

Instances of a specialization



3.2.2: Tổng quát hóa (Generalization):

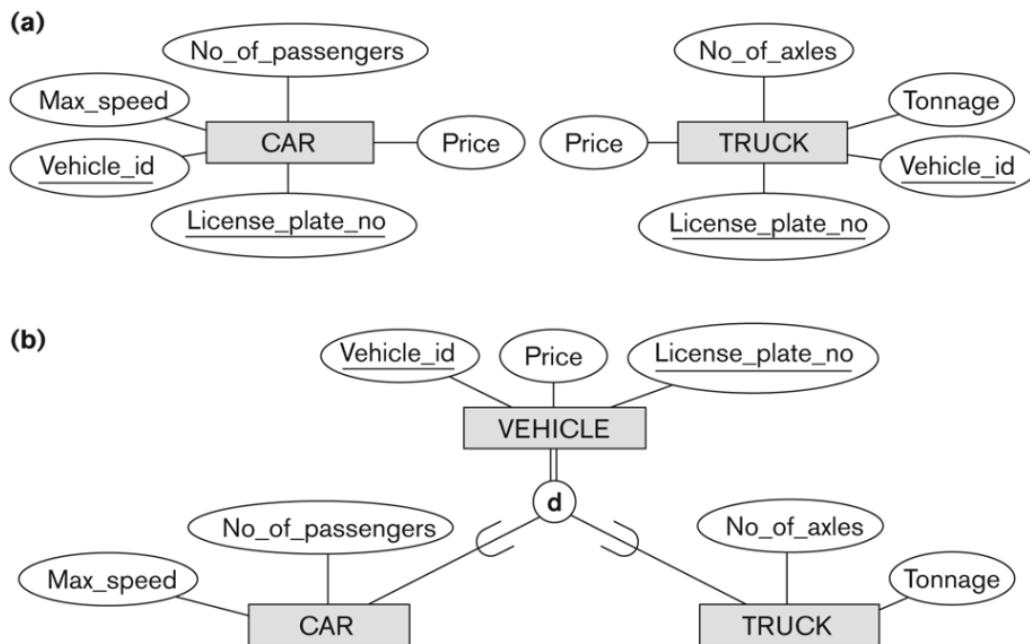
-Đồo ngược của quá trình Specialization.

-Tổng quát hóa các entity type mà có chung các feature thành một **superclass**.

-Coi các entity ban đầu là các lớp con đặc biệt.

*Định nghĩa: quá trình định nghĩa một kiểu thực thể tổng quát từ một tập kiểu thực thể.

Example of a Generalization



Generalization. (a) Two entity types, CAR and TRUCK.
(b) Generalizing CAR and TRUCK into the superclass VEHICLE.

3.2.3: Điều kiện ràng buộc (constraint) và đặc trưng của hệ thống phân cấp (hierarchy) cơ thể hóa và tổng quát hóa.

-Điều kiện ràng buộc được áp dụng cho một quá trình cơ thể hóa hoặc tổng quát hóa.

*Xác định subclass:

-Xác định theo vế t (hoặc được xác định theo điều kiện).

+Nếu chúng ta xác định chính xác được entity nào sẽ là thành viên của một subclass bằng điều kiện, thì subclass được gọi là subclass xác định theo vế t (xác định theo điều kiện).

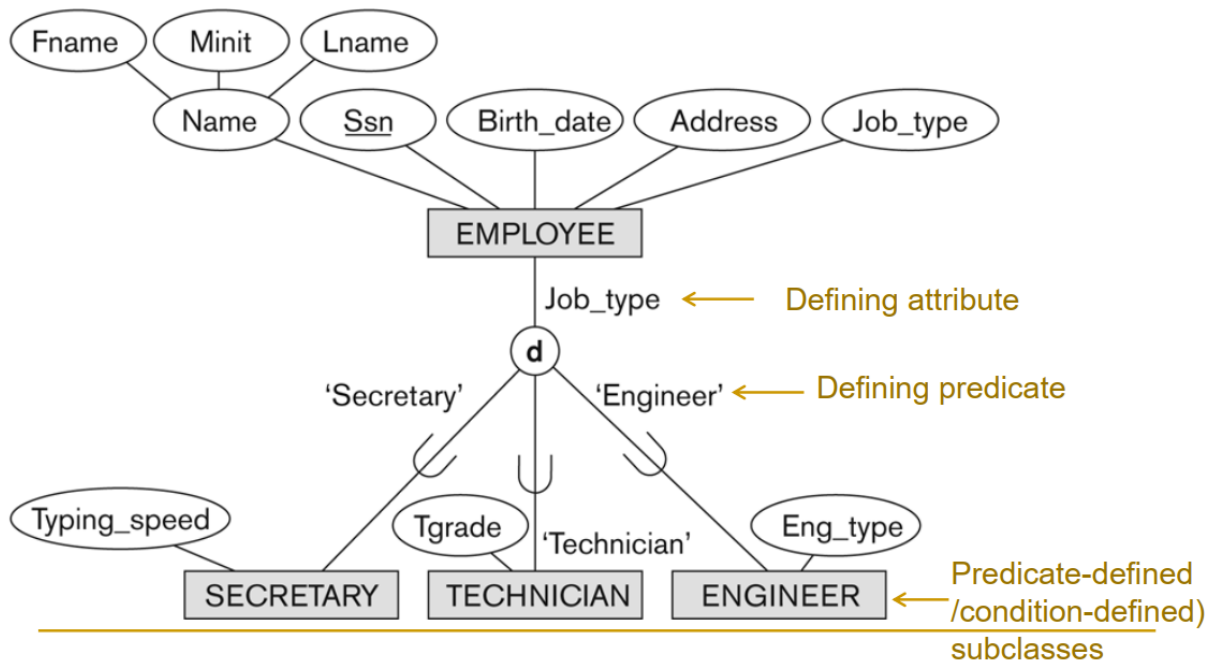
+Trường thái là điều kiện ràng buộc để xác định subclass member.

+Trình bày subclass xác định bằng điều kiện bằng cách ghi điều kiện vế t không bên line nối subclass với superclass.

-Cơ thể được xác định theo thuộc tính.

+ 'JobType' là thuộc tính định nghĩa của quá trình cụ thể hóa {SECRETARY, TECHNICIAN, ENGINEER}.

EER diagram notation for an attribute-defined specialization on JobType



-Người dùng định nghĩa.

+Nếu không có định nghĩa định nghĩa thì cách thành viên, subclass do người dùng định nghĩa.

+Tức cách thành viên trong subclass là xác định cho từng cá nhân cho mỗi entity trong superclass bởi user.

+Tức cách thành viên trong một subclass được định nghĩa bởi người dùng db bằng cách áp dụng một thao tác để thêm một **entity** vào **subclass**.

-Hai điều kiện áp dụng vào cụ thể hóa và tổng quát hóa là: **disjoints** (tính rời rạc), và **completeness** (tính toàn bộ).

+Disjoints: các subclass có được phép overlap (o) hay ko, nếu ko thì phải rời rạc (d). Được ký hiệu bằng vòng tròn bao quanh trong sơ đồ.

+Completeness: xác định toàn bộ các entity thuộc superclass có cần phân loại thành các loại subclass hay không. Kí hiệu đồ án thông 1 glich hoặc 2 glich nôi t superclass đn vòng tròn th hiên điu kien disjointness.

3.2.4.: Specialization and Generalization Hierarchy and Lattice:

-Specialization Hierarchy:

+Single inheritance.

+Tạo thành một cấu trúc cây hoặc k thỏa th bíc nghiêm ngặt.

-Specialization Lattice:

+Multiple inheritance.

-A subclass with more than one superclass is called **shared subclass**.

3.4: Category:

-Định nghĩa **Category** (loại) hoặc **Union** (hợp):

+Biểu diễn một quan hệ superclass/subclass đn với nhiều hơn 1 superclass.

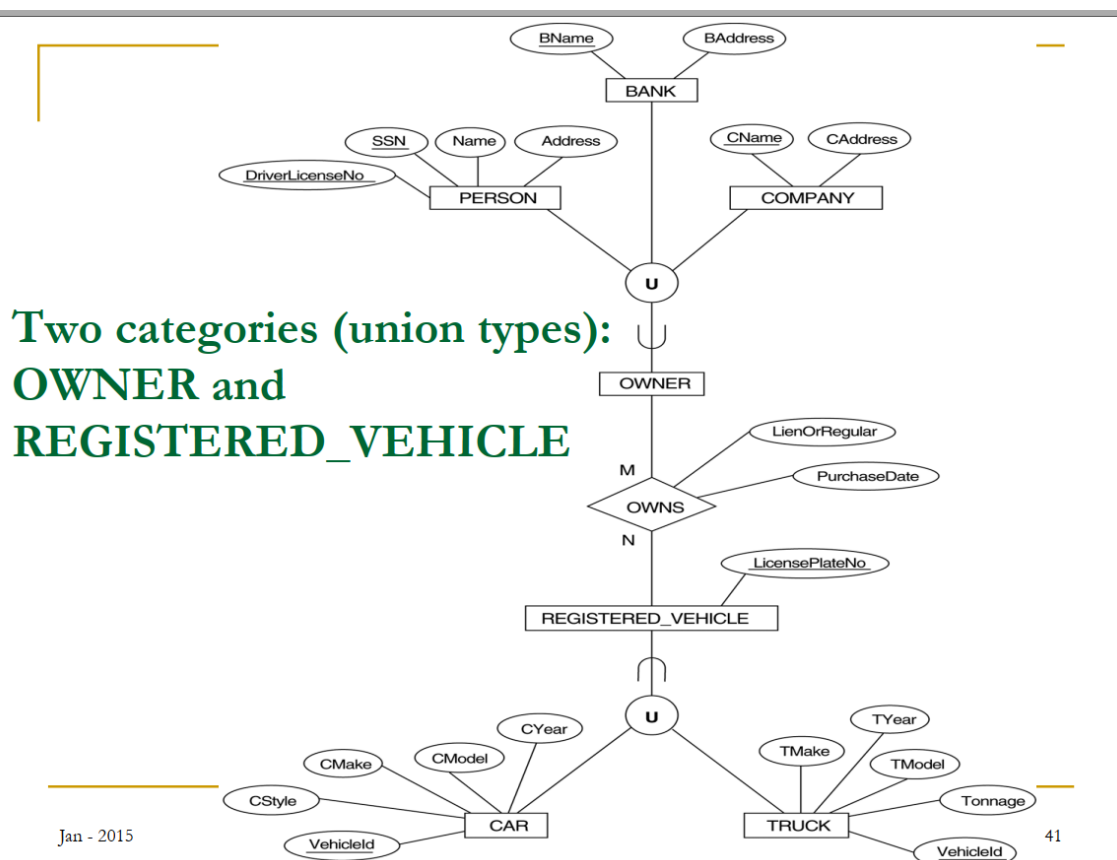
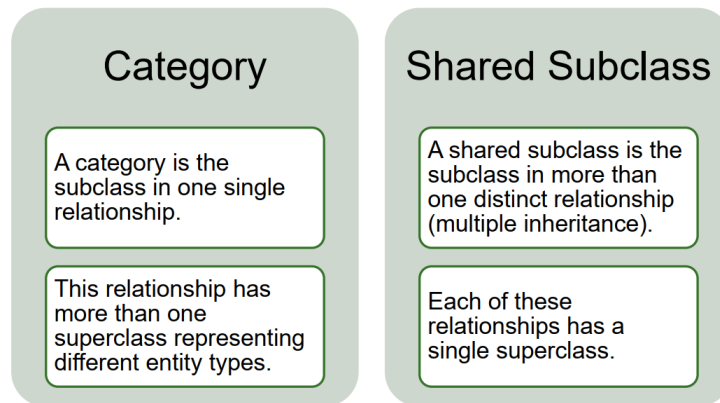
| Categories (2)

- Example: Database for vehicle registration, vehicle owner can be a person, a bank (holding a lien on a vehicle) or a company.
 - Category (subclass) OWNER is a subset of the union of the three superclasses COMPANY, BANK, and PERSON
 - A category member must exist in **only one** of its superclasses
- Note: The difference from shared subclass, which is a subset of the intersection of its superclasses (shared subclass member must exist in all of its superclasses)

-Ký hiệu: một chu bao bii đòng tròn, nôi gĩa hợp (union) các superclass với subclass trong quan hệ k thỏa đn.

-Subclass khi không thừa kế một union khác với shared subclass do chỉ chọn một thuộc tính từ 1 superclass nằm trong union, còn ở chiều ngược lại, shared subclass không thừa kế và chia tất cả các thuộc tính, quan hệ của các superclass mà nó không thừa kế.

Category vs. Shared Subclass (2)



3.5: Lựa chọn thiết kế cho specialization và generalization:

-Nếu subclass có ít attribute và không có quan hệ phức tạp: có thể merge với superclass.

-Loại union hoặc category nên tránh.

-Lựa chọn điều kiện ràng buộc để tránh disjoint/ overlapping và total/ partial trên specialization / generalization: được suy ra từ luật mà miniworld đang được mô hình hóa.

Formal Definitions for the EER Model Concepts

■ Subclass S :

- Inherits all the attributes and relationship of a class C
- Set of entities must always be a subset of the set of entities of the other class C: $S \subseteq C$
- C is called the superclass of S
- A superclass/subclass relationship exists between S and C

Formal Definitions for the EER Model Concepts

■ Class C

- Set or collection of entities
- Includes any of the EER schema constructs of group entities
- Can be entity type, subclass, superclass, or category
- Note: The definition of *relationship type* in ER/EER should have 'entity type' replaced with 'class' to allow relationships among classes in general

Formal Definitions for the EER Model Concepts

- **Specialization Z:** $Z = \{S1, S2, \dots, Sn\}$ is a set of subclasses with same superclass G;
 - G/S_i is a superclass/subclass relationship, $i=1..n$
 - G is called a generalization of the subclasses $\{S1, S2, \dots, Sn\}$
 - Z is total if:
 - $S1 \cup S2 \cup \dots \cup Sn = G$;
 - Otherwise, Z is partial
 - Z is disjoint if:
 - $S_i \cap S_j$ empty-set for $i \neq j$;
 - Otherwise, Z is overlapping

Formal Definitions for the EER Model Concepts

■ Generalization:

- Generalized entity type or superclass
- Subclass S of C is **predicate defined** if predicate (condition) p on attributes of C is used to specify membership in S; that is, $S = C[p]$, where $C[p]$ is the set of entities in C that satisfy condition p
- A subclass not defined by a predicate is called **user-defined**

