

Bài 02:

MÔ HÌNH THỰC THỂ KẾT HỢP



Nội dung

1. Mô hình cơ sở dữ liệu
 2. Giới thiệu
 2. Các thành phần cơ bản
 3. Lược đồ E/R
 4. Ví dụ
- Bài tập

1. Mô hình cơ sở dữ liệu

- ✓ Một mô hình cơ sở dữ liệu là một tập hợp các khái niệm dùng để biểu diễn các cấu trúc của cơ sở dữ liệu
- ✓ Một số mô hình CSDL còn có thêm một tập hợp các phép toán cơ bản để đặc tả các thao tác trên CSDL

Các loại mô hình cơ sở dữ liệu

- ✓ Các mô hình dữ liệu bậc cao hoặc mô hình dữ liệu mức quan niệm cung cấp các khái niệm gắn liền với cách cảm nhận dữ liệu của nhiều người sử dụng
- ✓ Các mô hình dữ liệu bậc thấp hoặc các mô hình dữ liệu vật lý cung cấp các khái niệm mô tả chi tiết về việc dữ liệu được lưu trữ trong máy tính như thế nào
- ✓ Các mô hình dữ liệu thể hiện (mô hình dữ liệu mức logic) cung cấp những khái niệm mà người sử dụng có thể hiểu được, đồng thời không xa với cách tổ chức dữ liệu bên trong máy tính
- ✓ Mô hình quan hệ là một trong các mô hình dữ liệu mức logic được sử dụng nhiều nhất hiện nay
- ✓ Một số mô hình dữ liệu khác: mô hình dữ liệu đồ thị (graph database model), mô hình dữ liệu địa lý (geographical database model)

Lược đồ và trạng thái cơ sở dữ liệu



- ✓ Mô tả của một cơ sở dữ liệu được gọi là lược đồ cơ sở dữ liệu, nó được xác định rõ trong quá trình thiết kế cơ sở dữ liệu và không bị thay đổi thường xuyên
- ✓ Các dữ liệu trong một cơ sở dữ liệu tại một thời điểm cụ thể được gọi là một trạng thái cơ sở dữ liệu hoặc là ảnh (snapshot) của cơ sở dữ liệu
- ✓ Trạng thái CSDL thường xuyên thay đổi trong quá trình sử dụng CSDL
- ✓ Cần phải phân biệt rõ giữa mô tả của cơ sở dữ liệu và bản thân cơ sở dữ liệu

Đặt vấn đề

- Theo dõi các thông tin liên quan tới nhân viên, phòng ban và đề án
- Công ty được tổ chức thành các phòng ban. Mỗi phòng ban có một tên duy nhất, mã số phòng ban duy nhất, và một nhân viên quản lý phòng đó. Ghi nhận lại ngày nhận chức trưởng phòng. Mỗi phòng ban có thể có nhiều địa điểm khác nhau
- Mỗi phòng ban chủ trì nhiều đề án, mỗi đề án có tên duy nhất, mã số đề án duy nhất và được triển khai ở một địa điểm
- Thông tin nhân viên của công ty được lưu trữ bao gồm mã nhân viên, địa chỉ, lương, phái, và ngày sinh. Mỗi nhân viên làm việc ở 1 phòng ban nhưng có thể tham gia nhiều đề án, trong đó đề án không nhất thiết phải do chính phòng ban của nhân viên chủ trì. Ghi nhận thông tin về thời gian tham gia đề án của nhân viên ứng với từng đề án tham gia, và cũng ghi nhận thông tin người quản lý trực tiếp nhân viên
- Mỗi nhân viên có thể có nhiều thân nhân, với mỗi thân nhân phải lưu trữ tên, phái, ngày sinh, và mối quan hệ với nhân viên trong công ty

Mô hình thực thể-liên kết

- ✓ Mô hình thực thể-liên kết (Entity-Relationship, viết tắt ER) là một mô hình dữ liệu mức quan niệm nhằm mô tả các đối tượng trong thế giới thực và quan hệ giữa chúng
- ✓ Thực thể là một đối tượng trong thế giới thực, có sự tồn tại độc lập:
 - Thực thể cụ thể: có thể cảm nhận bằng giác quan, ví dụ xe đạp, bàn, ghế
 - Thực thể trừu tượng: có thể nhận biết bằng nhận thức, ví dụ công ty, trường học



Thuộc tính (Attributes):

- Là những **đặc tính riêng biệt** của tập thực thể
- Tính chất của thực thể **cần được quản lý**
- Chỉ quan tâm đến các tính chất **có liên quan** đến ứng dụng

Ví dụ thực thể NHÂN VIÊN sẽ có những thuộc tính như Họ tên, Mã nhân viên, ngày sinh, địa chỉ, giới tính

Phân loại thuộc tính của thực thể



- **Thuộc tính đơn** là thuộc tính không thể phân chia ra được thành các thành phần nhỏ hơn
- **Thuộc tính phức hợp** là thuộc tính có thể phân chia được thành các thành phần nhỏ hơn, biểu diễn các thuộc tính cơ bản hơn với các ý nghĩa độc lập
- Những thuộc tính có giá trị duy nhất cho một thực thể cụ thể gọi là các **thuộc tính đơn trị**
- Một thuộc tính có thể có một tập giá trị cho cùng một thực thể: **thuộc tính đa trị**
- Thuộc tính có giá trị có thể tính được thông qua giá trị của các thuộc tính khác gọi là **thuộc tính suy diễn được**
- Trong một số trường hợp, một số thuộc tính của một thực thể cụ thể không xác định được giá trị. Trong trường hợp như vậy, ta phải tạo ra một giá trị đặc biệt gọi là giá trị null. Các thuộc tính nói trên là thuộc tính có thể nhận giá trị null

Thuộc tính của kiểu thực thể: định nghĩa hình thức

✓ Một thuộc tính A có tập giá trị V của kiểu thực thể E là một hàm

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

trong đó $P(V) = \{S, S \subseteq V\}$ là tập hợp tất cả các tập con của V .

$P(V)$ còn được gọi là tập lực lượng của V .

✓ Ký hiệu giá trị của thuộc tính A với thực thể e là $A(e)$

- Định nghĩa trên đúng cho các thuộc tính đơn, đa trị và thuộc tính có thể nhận giá trị null
- Với các thuộc tính đơn trị, $A(e)$ có duy nhất một phần tử
- Với các thuộc tính có thể nhận giá trị null, $A(e)$ là tập rỗng

Kiểu thực thể và tập thực thể



- ✓ Một kiểu thực thể là một nhóm các thực thể có các thuộc tính như nhau được mô tả bằng tên và các thuộc tính.
 - Ví dụ: NHÂNVIÊN (Họ tên, Tuổi, Lương) là một kiểu thực thể
- ✓ Một tập hợp các thực thể của một kiểu thực thể trong cơ sở dữ liệu tại một thời điểm bất kỳ được gọi là một tập thực thể

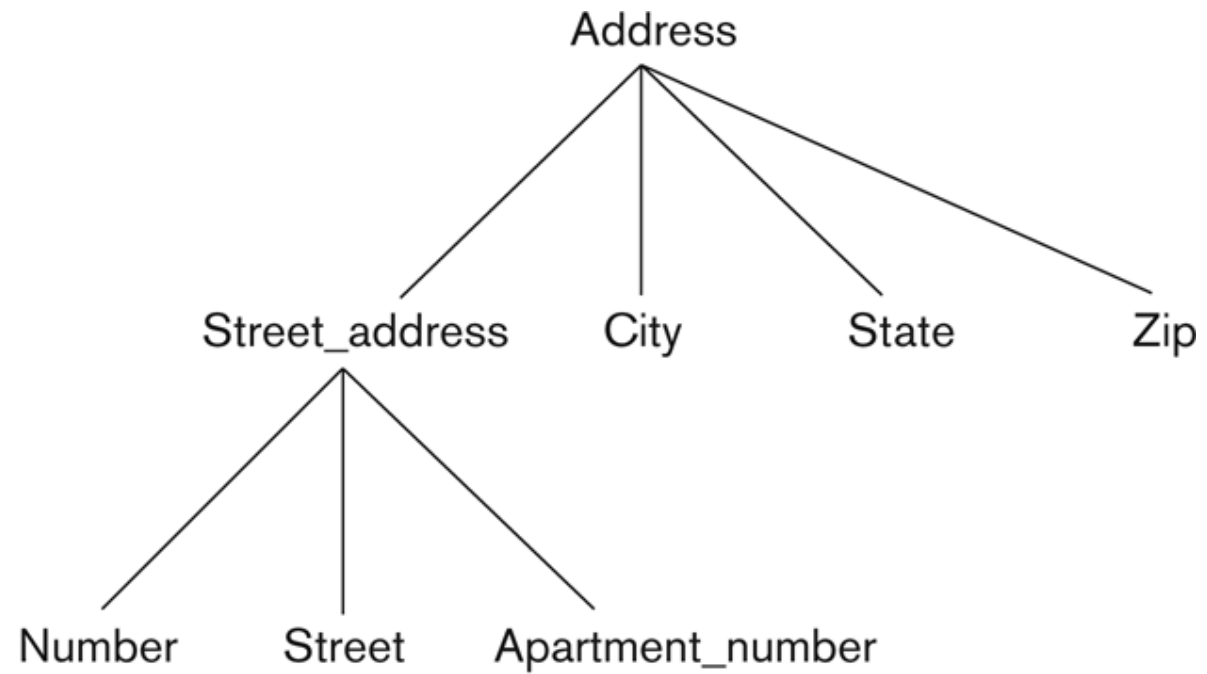


Figure 3.4
A hierarchy of
composite attributes.

Khóa và tập giá trị

- ✓ Thuộc tính mà các giá trị của nó là khác nhau đối với mỗi thực thể riêng biệt trong một tập thực thể gọi là **thuộc tính khóa** → khóa dùng để phân biệt hai thực thể
- ✓ Nhiều thuộc tính kết hợp với nhau tạo thành một **khóa phức hợp**. Khóa phức hợp phải *tối thiểu*
- ✓ Một kiểu thực thể có thể có nhiều hơn một khóa
- ✓ Kiểu thực thể không có khóa gọi là **kiểu thực thể yếu**
- ✓ Mỗi thuộc tính đơn của một kiểu thực thể được kết hợp với một **miền giá trị**

Entity Types and Key Attributes (2)

- ✓ A key attribute may be composite.
 - VehicleTagNumber is a key of the CAR entity type with components (Number, State).
- ✓ An entity type may have more than one key.
 - The CAR entity type may have two keys:
 - VehicleIdentificationNumber (popularly called VIN)
 - VehicleTagNumber (Number, State), aka license plate number.
- ✓ Each key is underlined

(a)

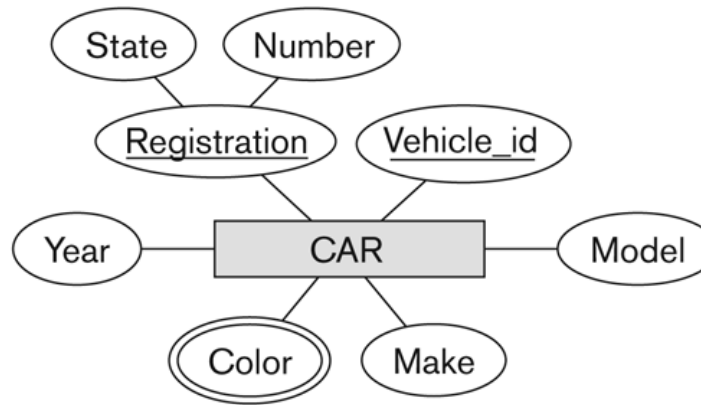


Figure 3.7

The CAR entity type with two key attributes, Registration and Vehicle_id. (a) ER diagram notation. (b) Entity set with three entities.

(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})

⋮

Ví dụ 2

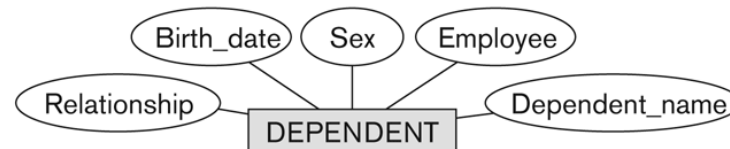
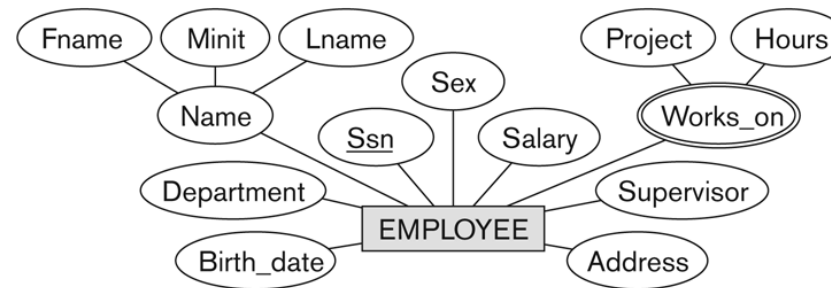
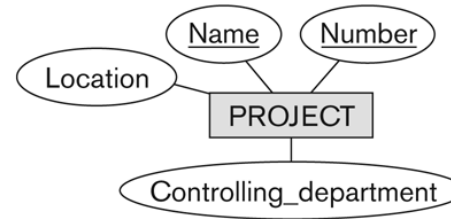
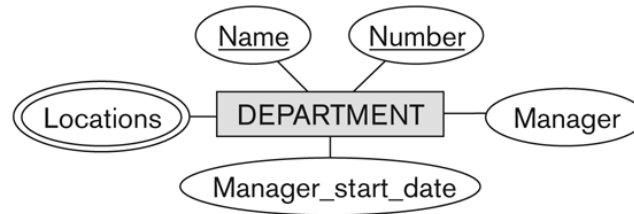


Figure 3.8
Preliminary design of entity types for the COMPANY database. Some of the shown attributes will be refined into relationships.

Mối kết hợp



- ✓ Diễn tả mối quan hệ ngữ nghĩa giữa ít nhất 2 thực thể khác nhau
- ✓ Biểu diễn quan hệ kết hợp giữa 2 thực thể gọi là mối kết hợp nhị phân.
- ✓ Biểu diễn quan hệ kết hợp giữa nhiều thực thể (từ 3 trở lên) gọi là mối kết hợp đa phân.
- ✓ Có tên gọi riêng
- ✓ Có thuộc tính riêng của mối kết hợp
- ✓ Khóa hiển nhiên: không cần mô tả → được suy từ mô tả mối kết hợp (tổ hợp khóa)

Relationships and Relationship Types (1)

- ✓ A **relationship** relates two or more distinct entities with a specific meaning.
 - For example, EMPLOYEE John Smith *works on* the ProductX PROJECT, or EMPLOYEE Franklin Wong *manages* the Research DEPARTMENT.
- ✓ Relationships of the same type are grouped or typed into a **relationship type**.
 - For example, the WORKS_ON relationship type in which EMPLOYEEs and PROJECTs participate, or the MANAGES relationship type in which EMPLOYEEs and DEPARTMENTs participate.
- ✓ The degree of a relationship type is the number of participating entity types.
 - Both MANAGES and WORKS_ON are *binary* relationships.

VD: Mỗi kết hợp giữa EMPLOYEE và DEPARTMENT

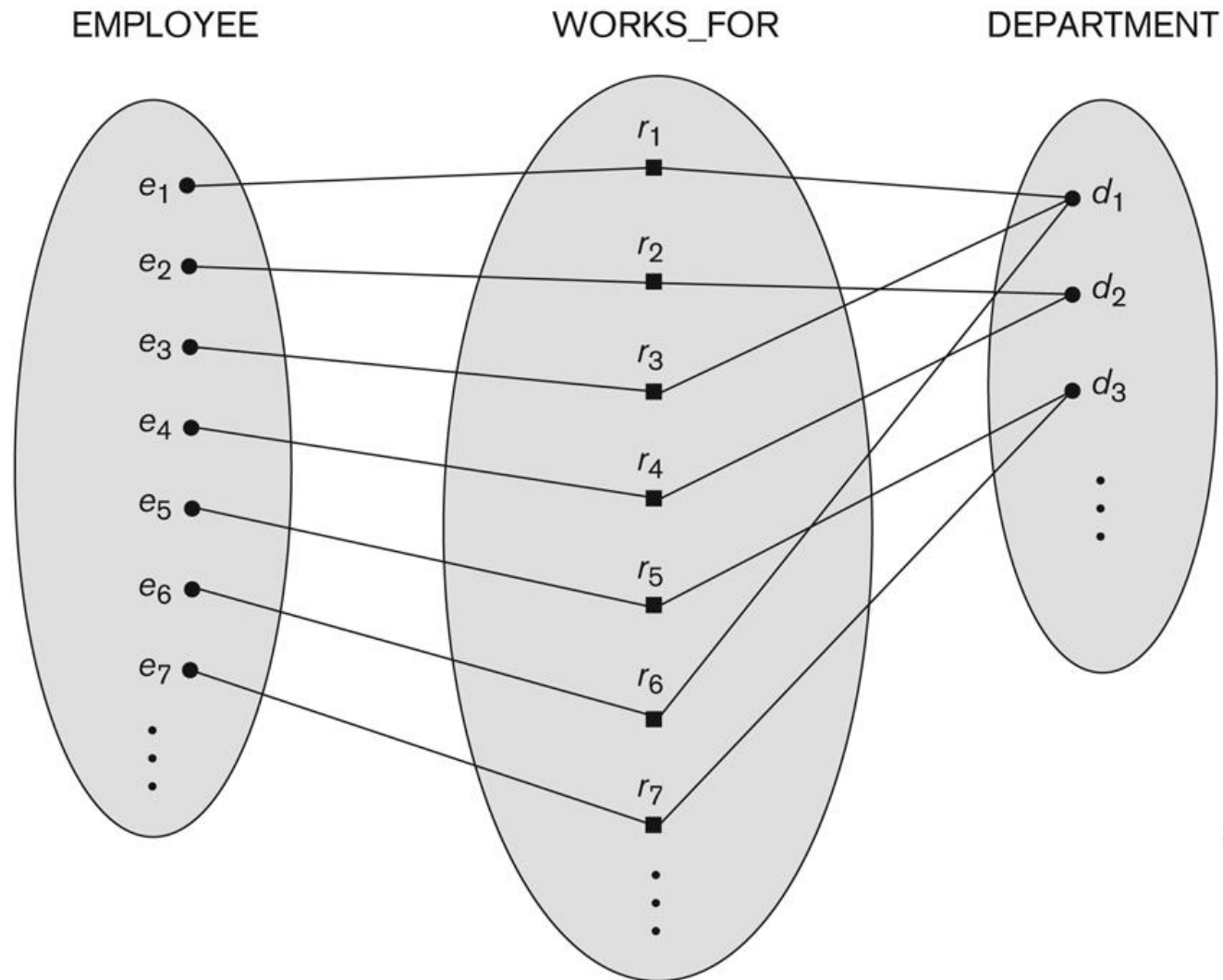


Figure 3.9

Some instances in the WORKS_FOR relationship set, which represents a relationship type WORKS_FOR between EMPLOYEE and DEPARTMENT.

Mối kết hợp WORKS_ON với EMPLOYEE và PROJECT

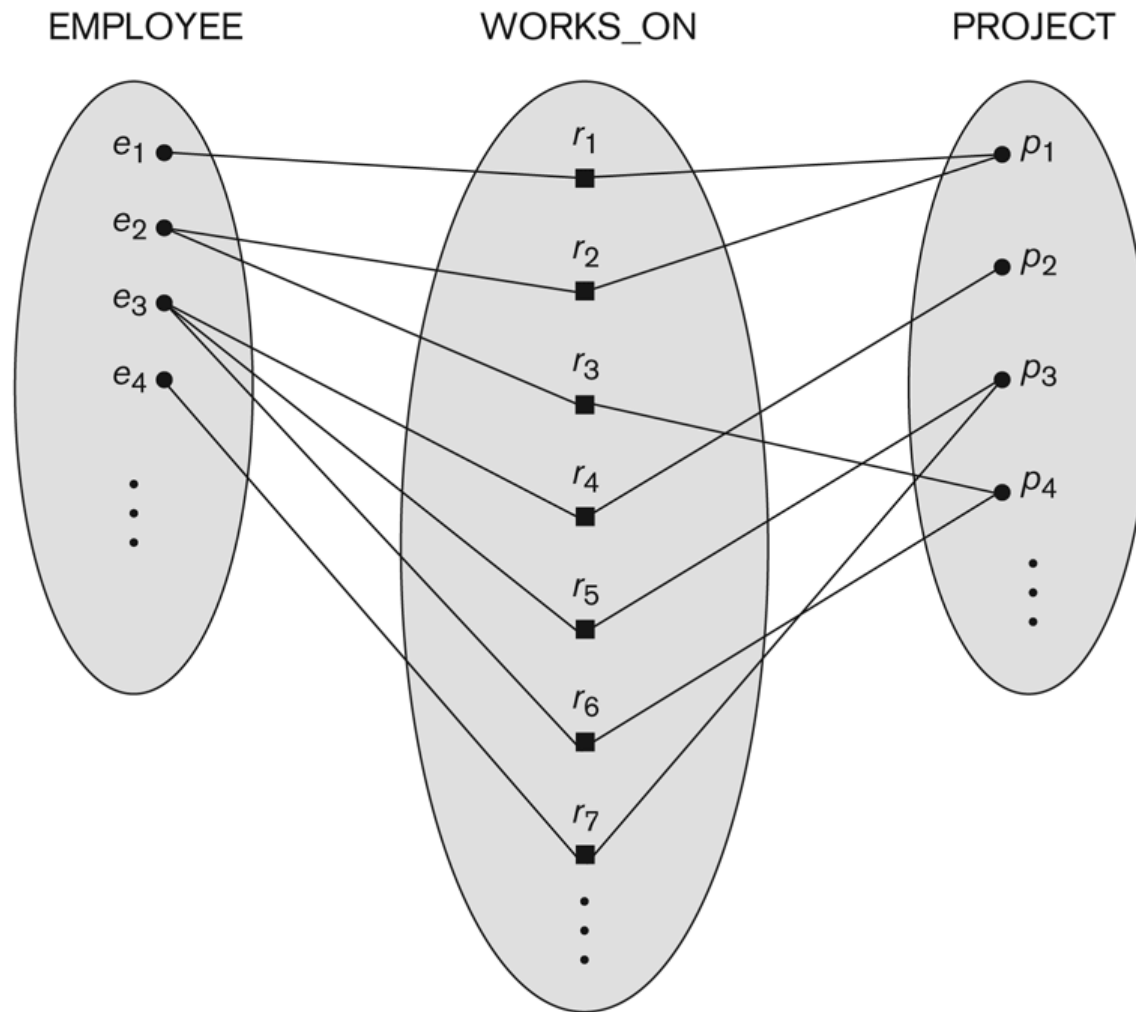


Figure 3.13
An M:N relationship,
WORKS_ON.

Relationship type vs. relationship set (1)

✓ Relationship Type:

- Is the schema description of a relationship
- Identifies the relationship name and the participating entity types
- Also identifies certain relationship constraints

✓ Relationship Set:

- The current set of relationship instances represented in the database
- The current *state* of a relationship type

Relationship type vs. relationship set (2)



- ✓ Previous figures displayed the relationship sets
- ✓ Each instance in the set relates individual participating entities – one from each participating entity type
- ✓ In ER diagrams, we represent the *relationship type* as follows:
 - Diamond-shaped box is used to display a relationship type
 - Connected to the participating entity types via straight lines

Refining the COMPANY database schema by introducing relationships

- ✓ By examining the requirements, six relationship types are identified
- ✓ All are *binary* relationships(degree 2)
- ✓ Listed below with their participating entity types:
 - WORKS_FOR (between EMPLOYEE, DEPARTMENT)
 - MANAGES (also between EMPLOYEE, DEPARTMENT)
 - CONTROLS (between DEPARTMENT, PROJECT)
 - WORKS_ON (between EMPLOYEE, PROJECT)
 - SUPERVISION (between EMPLOYEE (as subordinate), EMPLOYEE (as supervisor))
 - DEPENDENTS_OF (between EMPLOYEE, DEPENDENT)

ER DIAGRAM – Relationship Types are:

WORKS_FOR, MANAGES, WORKS_ON, CONTROLS, SUPERVISION, DEPENDENTS_OF

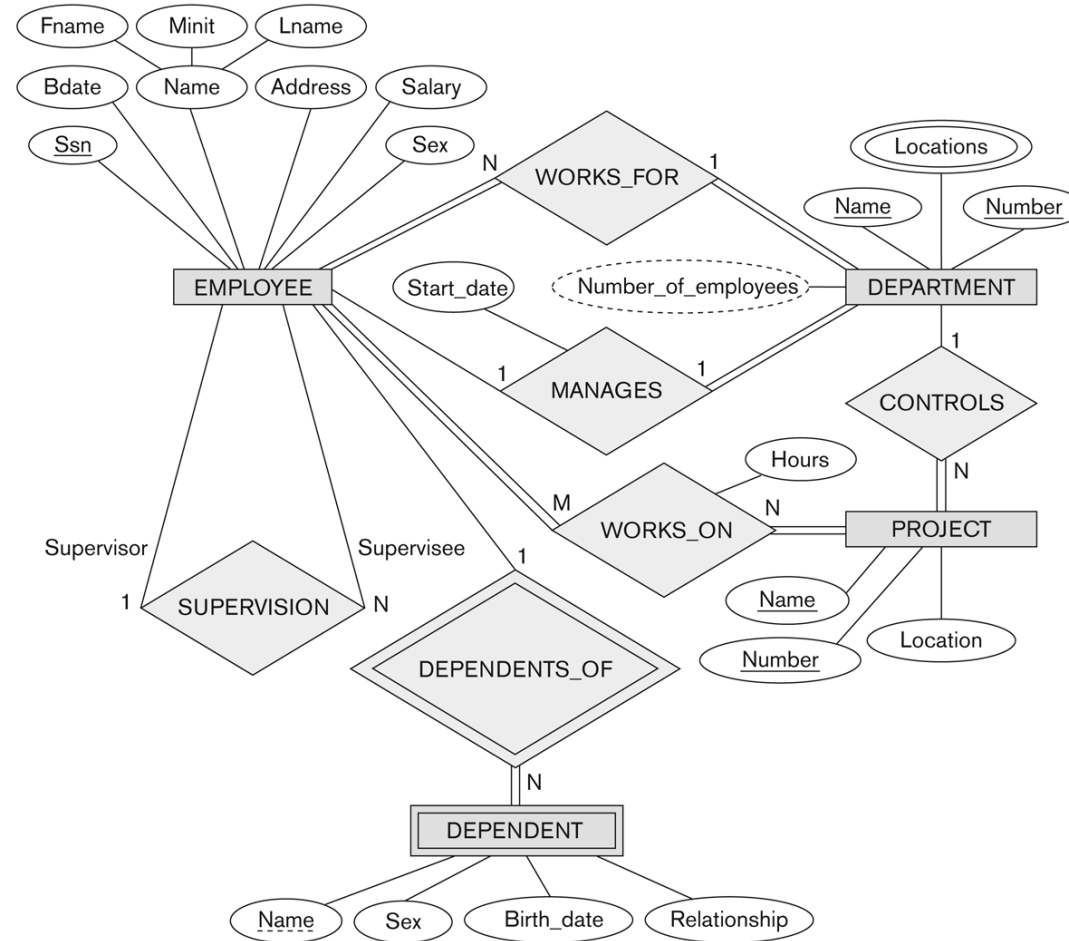


Figure 3.2

An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

Discussion on Relationship Types



- ✓ In the refined design, some attributes from the initial entity types are refined into relationships:
 - Manager of DEPARTMENT -> MANAGES
 - Works_on of EMPLOYEE -> WORKS_ON
 - Department of EMPLOYEE -> WORKS_FOR
 - etc
- ✓ In general, more than one relationship type can exist between the same participating entity types
 - MANAGES and WORKS_FOR are distinct relationship types between EMPLOYEE and DEPARTMENT
 - Different meanings and different relationship instances.

Recursive Relationship Type



- ✓ An relationship type whose with the same participating entity type in **distinct roles**
- ✓ Example: the SUPERVISION relationship
- ✓ EMPLOYEE participates twice in two distinct roles:
 - supervisor (or boss) role
 - supervisee (or subordinate) role
- ✓ Each relationship instance relates two distinct EMPLOYEE entities:
 - One employee in *supervisor* role
 - One employee in *supervisee* role

Weak Entity Types

- ✓ An entity that does not have a key attribute
- ✓ A weak entity must participate in an identifying relationship type with an owner or identifying entity type
- ✓ Entities are identified by the combination of:
 - A partial key of the weak entity type
 - The particular entity they are related to in the identifying entity type
- ✓ **Example:**
 - A DEPENDENT entity is identified by the dependent's first name, *and* the specific EMPLOYEE with whom the dependent is related
 - Name of DEPENDENT is the *partial key*
 - DEPENDENT is a *weak entity type*
 - EMPLOYEE is its identifying entity type via the identifying relationship type DEPENDENT_OF

Constraints on Relationships



✓ Constraints on Relationship Types

- (Also known as ratio constraints)
- Cardinality Ratio (specifies *maximum* participation)
 - One-to-one (1:1)
 - One-to-many (1:N) or Many-to-one (N:1)
 - Many-to-many (M:N)
- Existence Dependency Constraint (specifies *minimum* participation) (also called participation constraint)
 - zero (optional participation, not existence-dependent)
 - one or more (mandatory participation, existence-dependent)

Many-to-one (N:1) Relationship

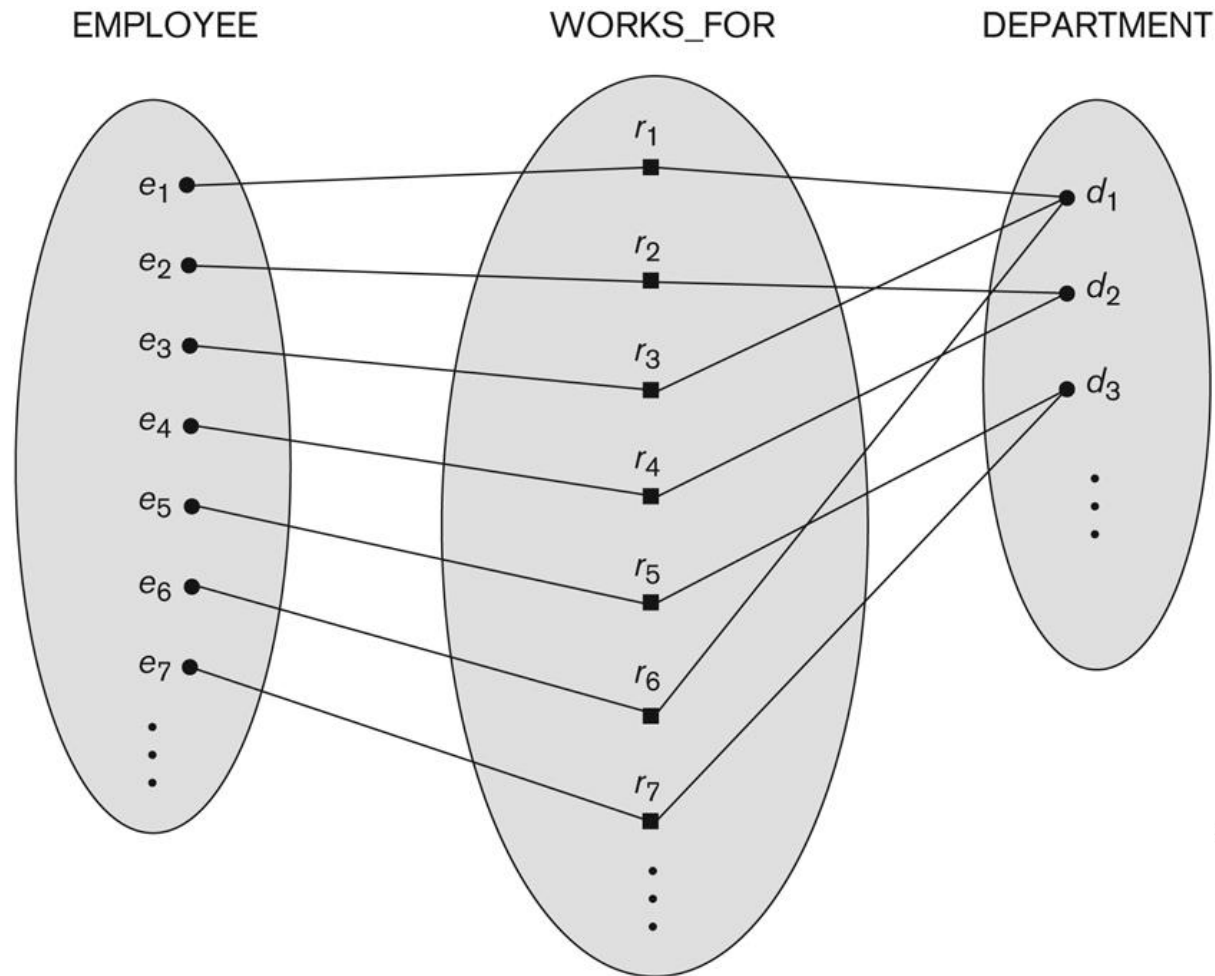


Figure 3.9

Some instances in the WORKS_FOR relationship set, which represents a relationship type WORKS_FOR between EMPLOYEE and DEPARTMENT.

Many-to-many (M:N) Relationship

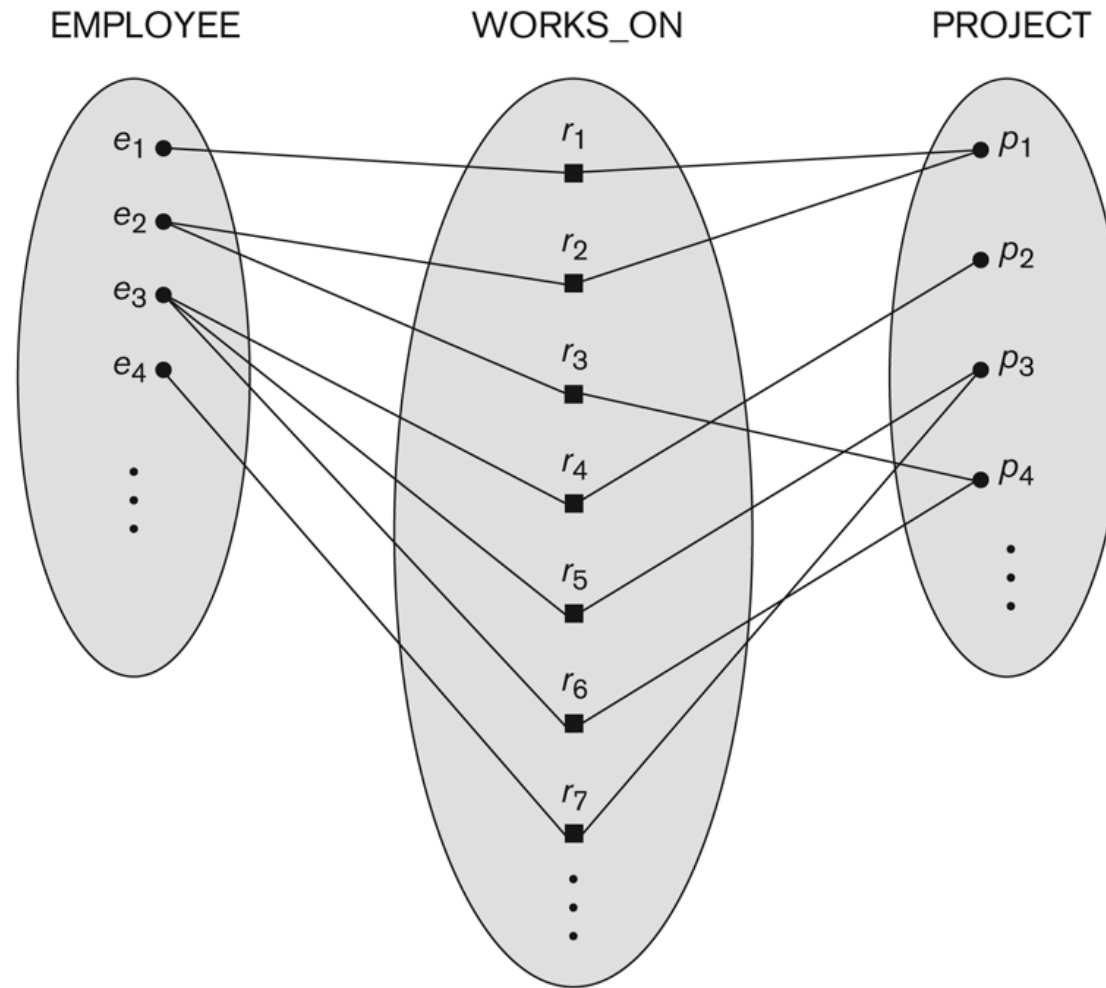


Figure 3.13
An M:N relationship,
WORKS_ON.

Displaying a recursive relationship



- ✓ In a recursive relationship type.
 - Both participations are same entity type in different roles.
 - For example, SUPERVISION relationships between EMPLOYEE (in role of supervisor or boss) and (another) EMPLOYEE (in role of subordinate or worker).
- ✓ In following figure, first role participation labeled with 1 and second role participation labeled with 2.
- ✓ In ER diagram, need to display role names to distinguish participations.

A Recursive Relationship Supervision`

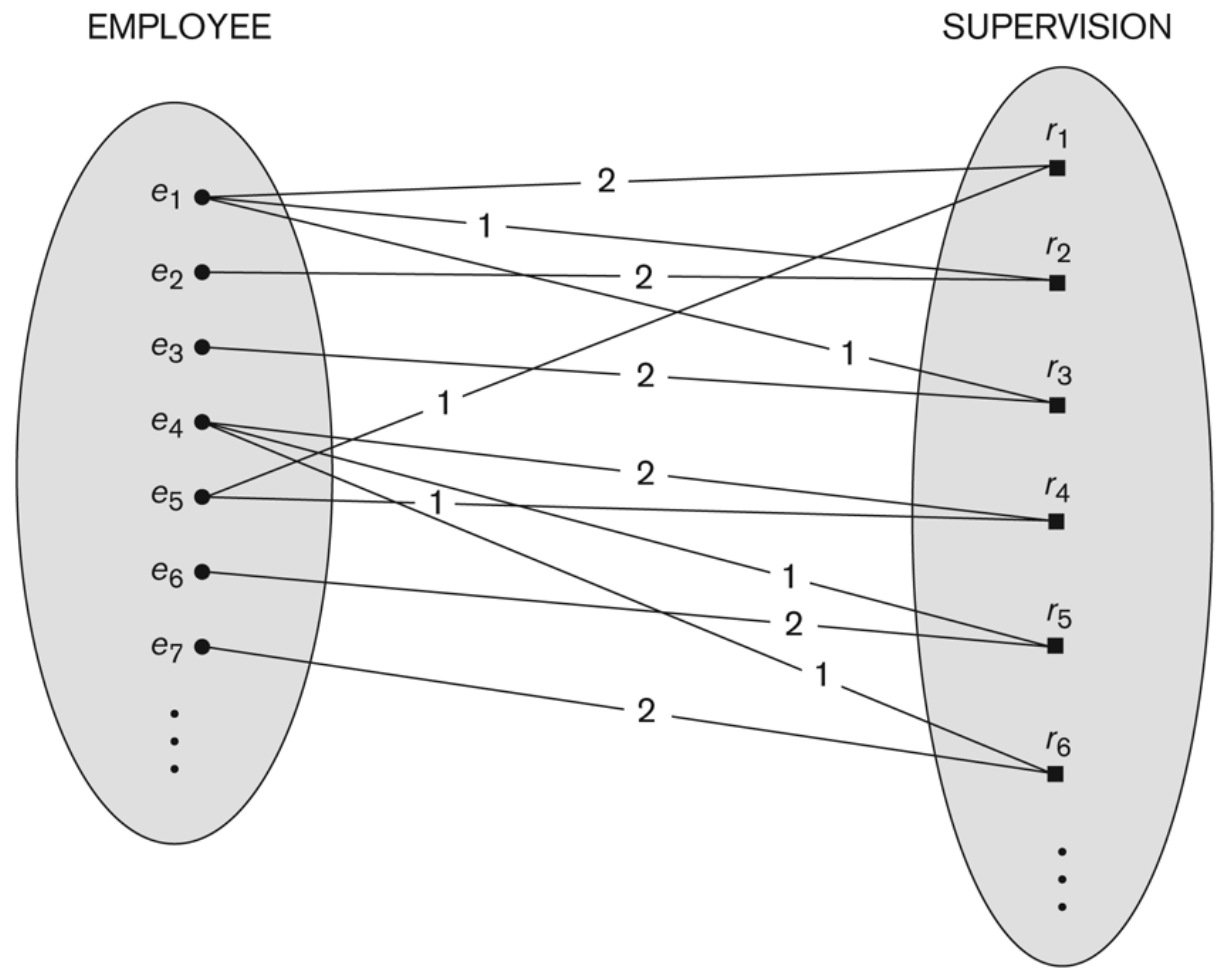


Figure 3.11
A recursive relationship SUPERVISION between EMPLOYEE in the *supervisor* role (1) and EMPLOYEE in the *subordinate* role (2).

Recursive Relationship Type is: SUPERVISION (participation role names are shown)

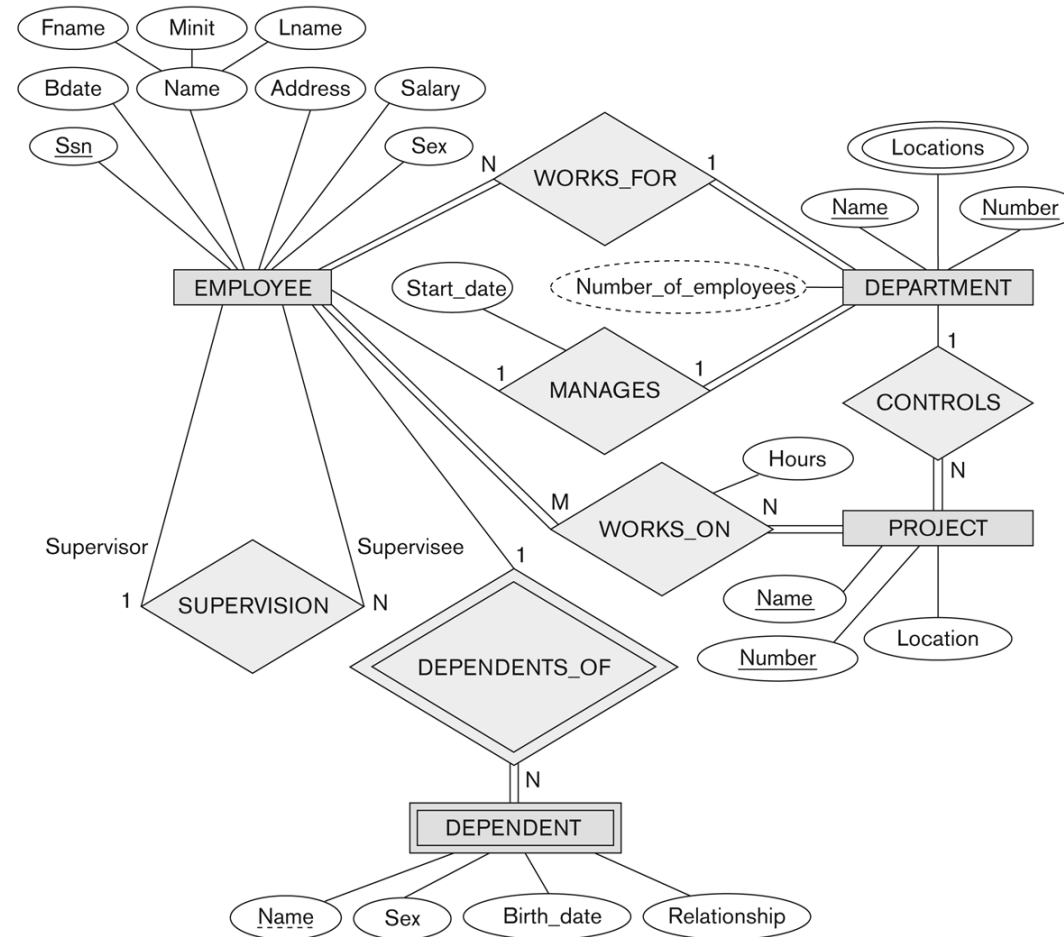


Figure 3.2

An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

Attributes of Relationship types



- ✓ A relationship type can have attributes:
 - For example, HoursPerWeek of WORKS_ON
 - Its value for each relationship instance describes the number of hours per week that an EMPLOYEE works on a PROJECT.
 - A value of HoursPerWeek depends on a particular (employee, project) combination
 - Most relationship attributes are used with M:N relationships
 - In 1:N relationships, they can be transferred to the entity type on the N-side of the relationship

Example Attribute of a Relationship Type: Hours of WORKS_ON

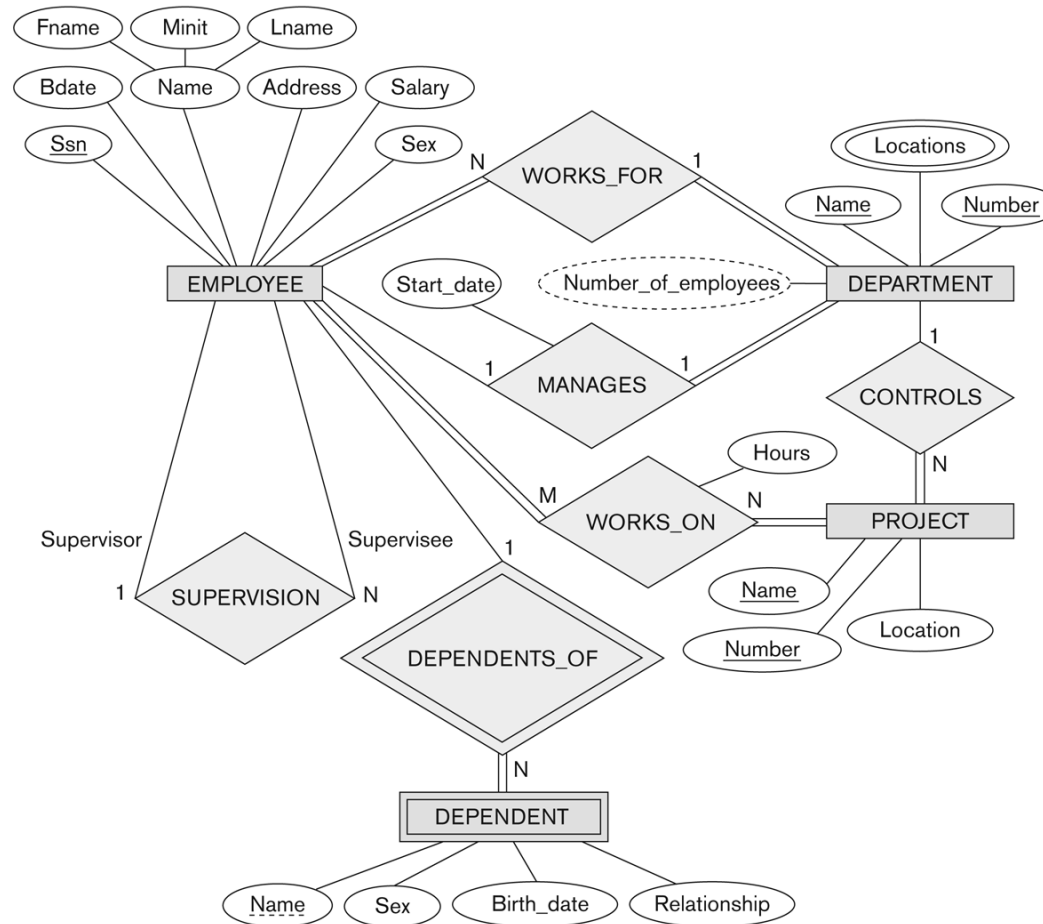


Figure 3.2

An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

Notation for Constraints on Relationships

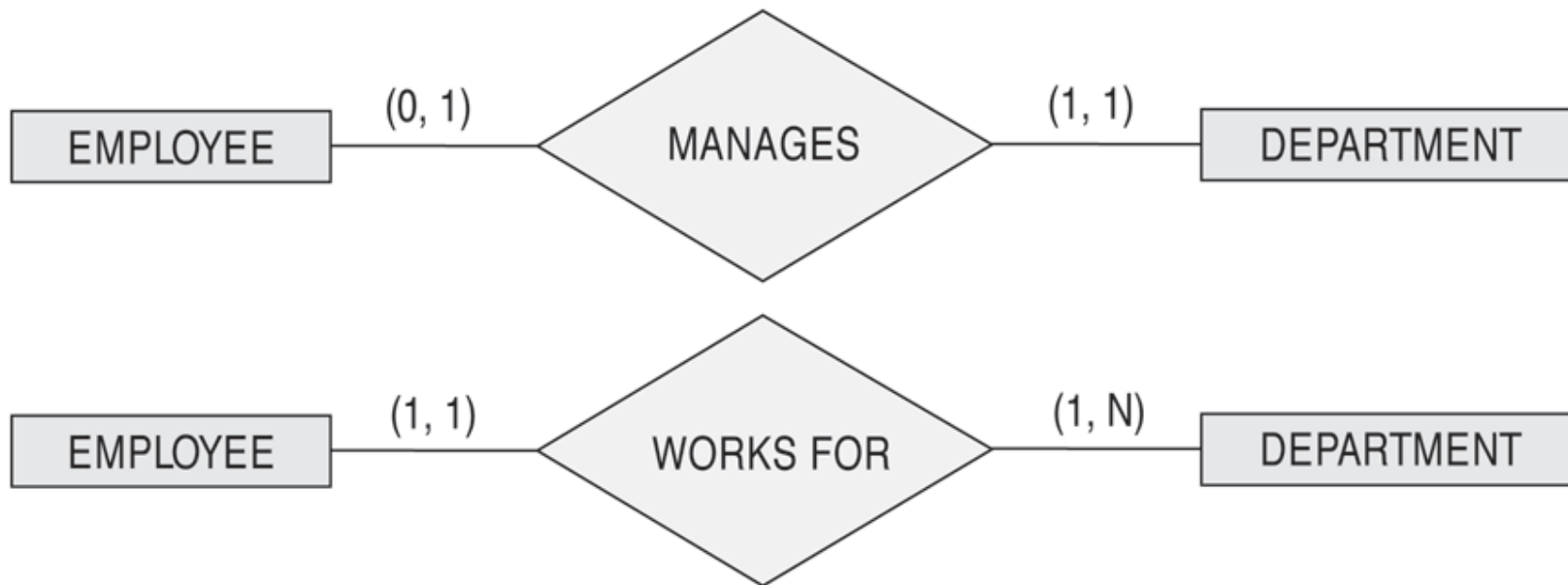


- ✓ Cardinality ratio (of a binary relationship): 1:1, 1:N, N:1, or M:N
 - Shown by placing appropriate numbers on the relationship edges.
- ✓ Participation constraint (on each participating entity type): total (called existence dependency) or partial.
 - Total shown by double line, partial by single line.
- ✓ NOTE: These are easy to specify for Binary Relationship Types.

Alternative (min, max) notation for relationship structural constraints:

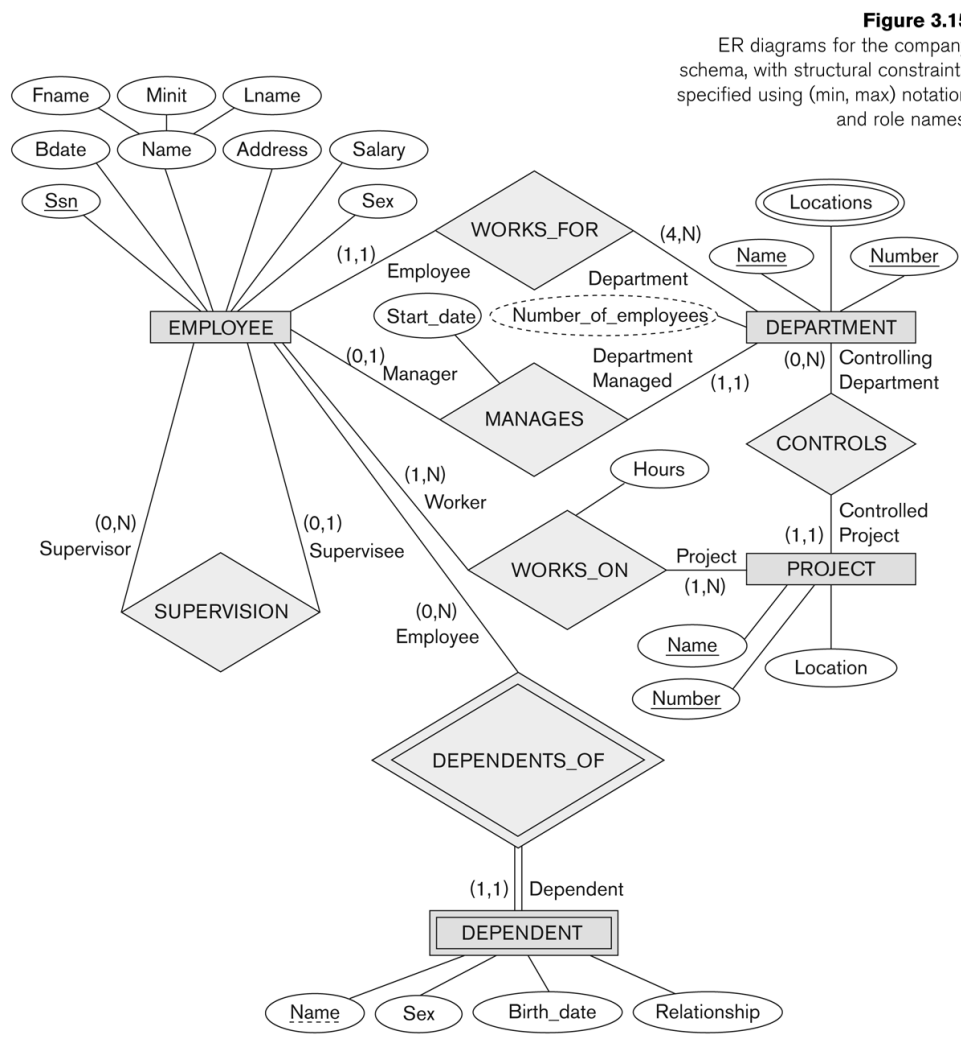
- ✓ Specified on each participation of an entity type E in a relationship type R
- ✓ Specifies that each entity e in E participates in at least *min* and at most *max* relationship instances in R
- ✓ Default(no constraint): min=0, max=n (signifying no limit)
- ✓ Must have $\text{min} \leq \text{max}$, $\text{min} \geq 0$, $\text{max} \geq 1$
- ✓ Derived from the knowledge of mini-world constraints
- ✓ Examples:
 - A department has exactly one manager and an employee can manage at most one department.
 - Specify (0,1) for participation of EMPLOYEE in MANAGES
 - Specify (1,1) for participation of DEPARTMENT in MANAGES
 - An employee can work for exactly one department but a department can have any number of employees.
 - Specify (1,1) for participation of EMPLOYEE in WORKS_FOR
 - Specify (0,n) for participation of DEPARTMENT in WORKS_FOR

The (min,max) notation for relationship constraints



Read the min,max numbers next to the entity type and looking **away from** the entity type

COMPANY ER Schema Diagram using (min, max) notation




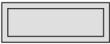





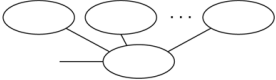

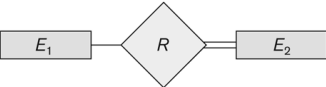

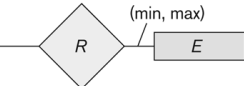
Alternative diagrammatic notation



- ✓ ER diagrams is one popular example for displaying database schemas
- ✓ Many other notations exist in the literature and in various database design and modeling tools
- ✓ Appendix A illustrates some of the alternative notations that have been used
- ✓ UML class diagrams is representative of another way of displaying ER concepts that is used in several commercial design tools

Summary of notation for ER diagrams

Figure 3.14
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

UML class diagrams

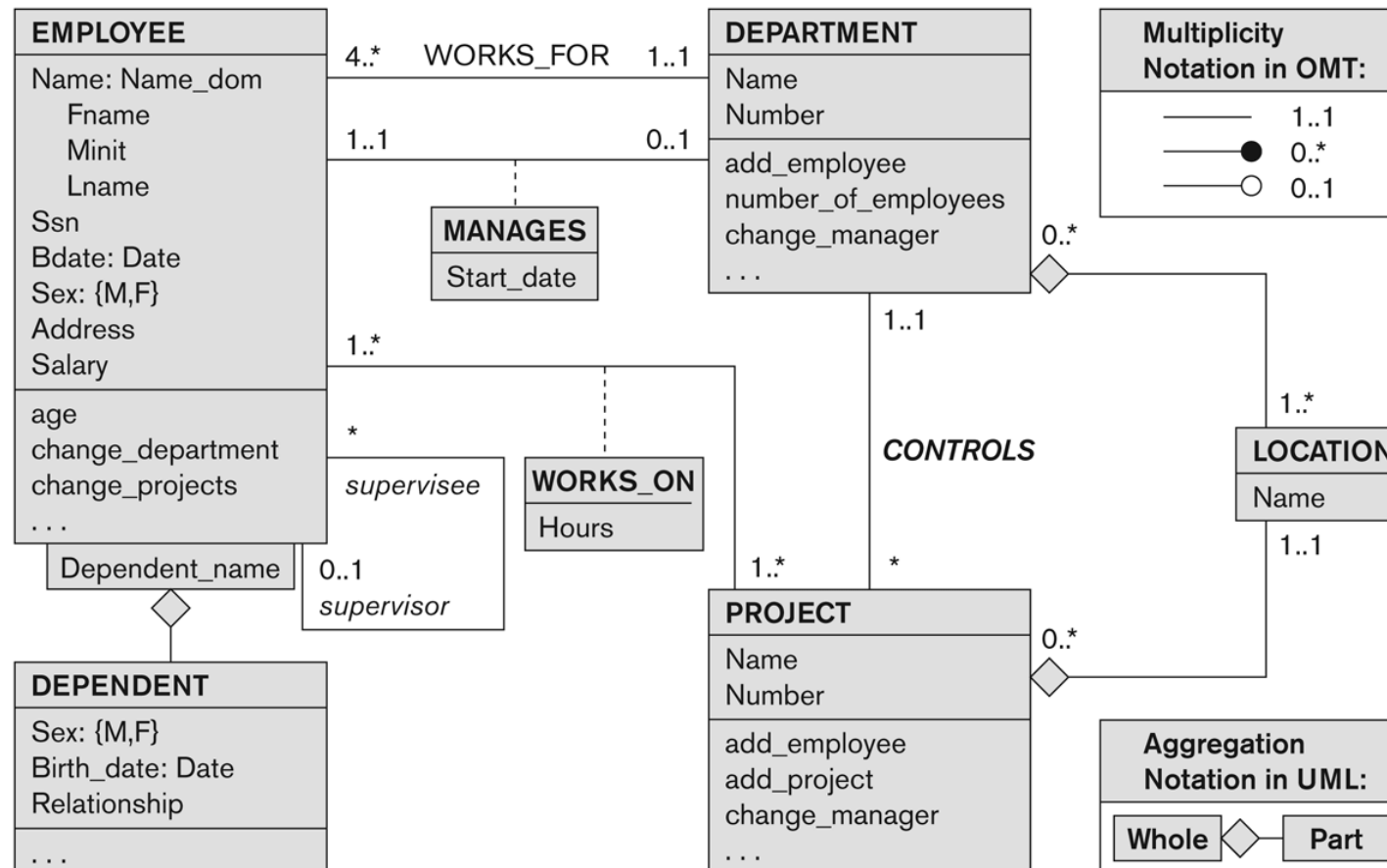


- ✓ Represent classes (similar to entity types) as large rounded boxes with three sections:
 - Top section includes entity type (class) name
 - Second section includes attributes
 - Third section includes class operations (operations are not in basic ER model)
- ✓ Relationships (called associations) represented as lines connecting the classes
 - Other UML terminology also differs from ER terminology
- ✓ Used in database design and object-oriented software design
- ✓ UML has many other types of diagrams for software design (see Chapter 12)

UML class diagram for COMPANY database schema

Figure 3.16

The COMPANY conceptual schema in UML class diagram notation.



Other alternative diagrammatic notations

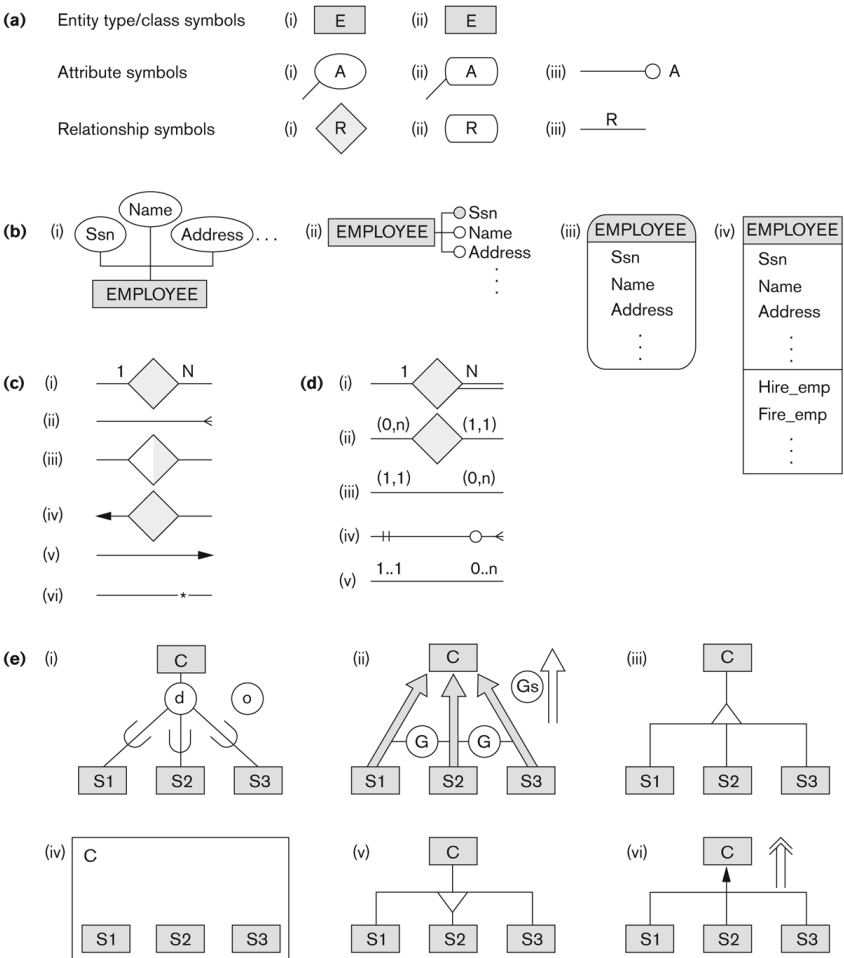


Figure A.1
Alternative notations. (a) Symbols for entity type/class, attribute, and relationship. (b) Displaying attributes. (c) Displaying cardinality ratios. (d) Various (min, max) notations. (e) Notations for displaying specialization/generalization.

Relationships of Higher Degree



- ✓ Relationship types of degree 2 are called binary
- ✓ Relationship types of degree 3 are called ternary and of degree n are called n -ary
- ✓ In general, an n -ary relationship is not equivalent to n binary relationships
- ✓ Constraints are harder to specify for higher-degree relationships ($n > 2$) than for binary relationships

Discussion of n-ary relationships ($n > 2$)



- ✓ In general, 3 binary relationships can represent different information than a single ternary relationship (see Figure 3.17a and b on next slide)
- ✓ If needed, the binary and n-ary relationships can all be included in the schema design (see Figure 3.17a and b, where all relationships convey different meanings)
- ✓ In some cases, a ternary relationship can be represented as a weak entity if the data model allows a weak entity type to have multiple identifying relationships (and hence multiple owner entity types) (see Figure 3.17c)

Example of a ternary relationship

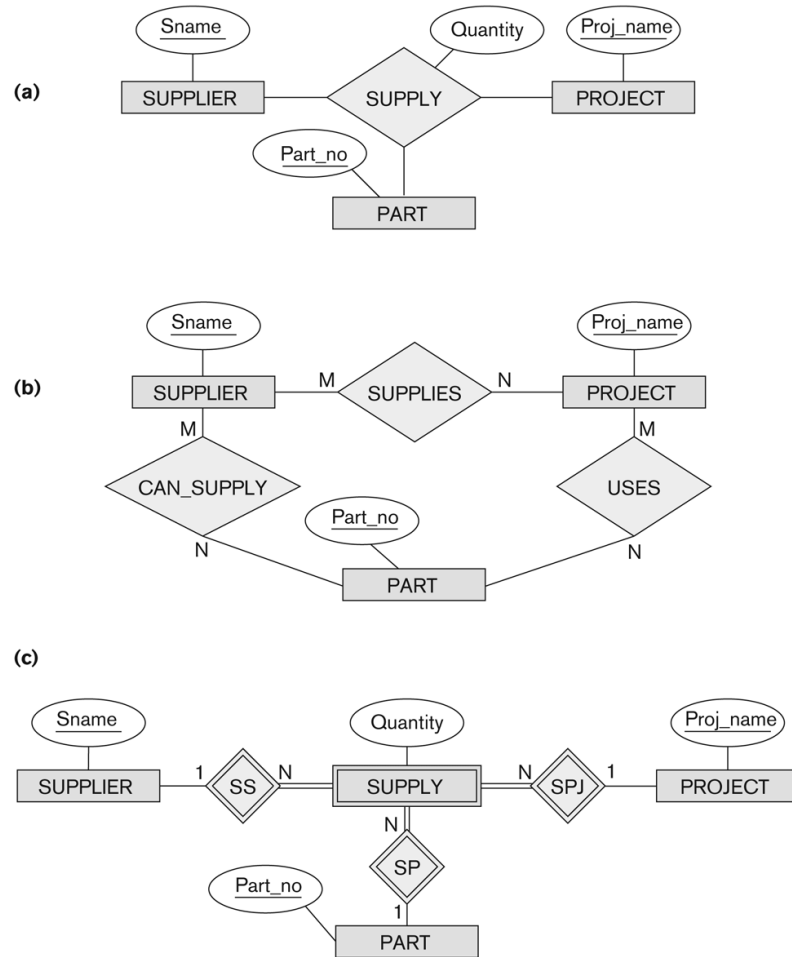


Figure 3.17

Ternary relationship types. (a) The **SUPPLY** relationship. (b) Three binary relationships not equivalent to **SUPPLY**. (c) **SUPPLY** represented as a weak entity type.

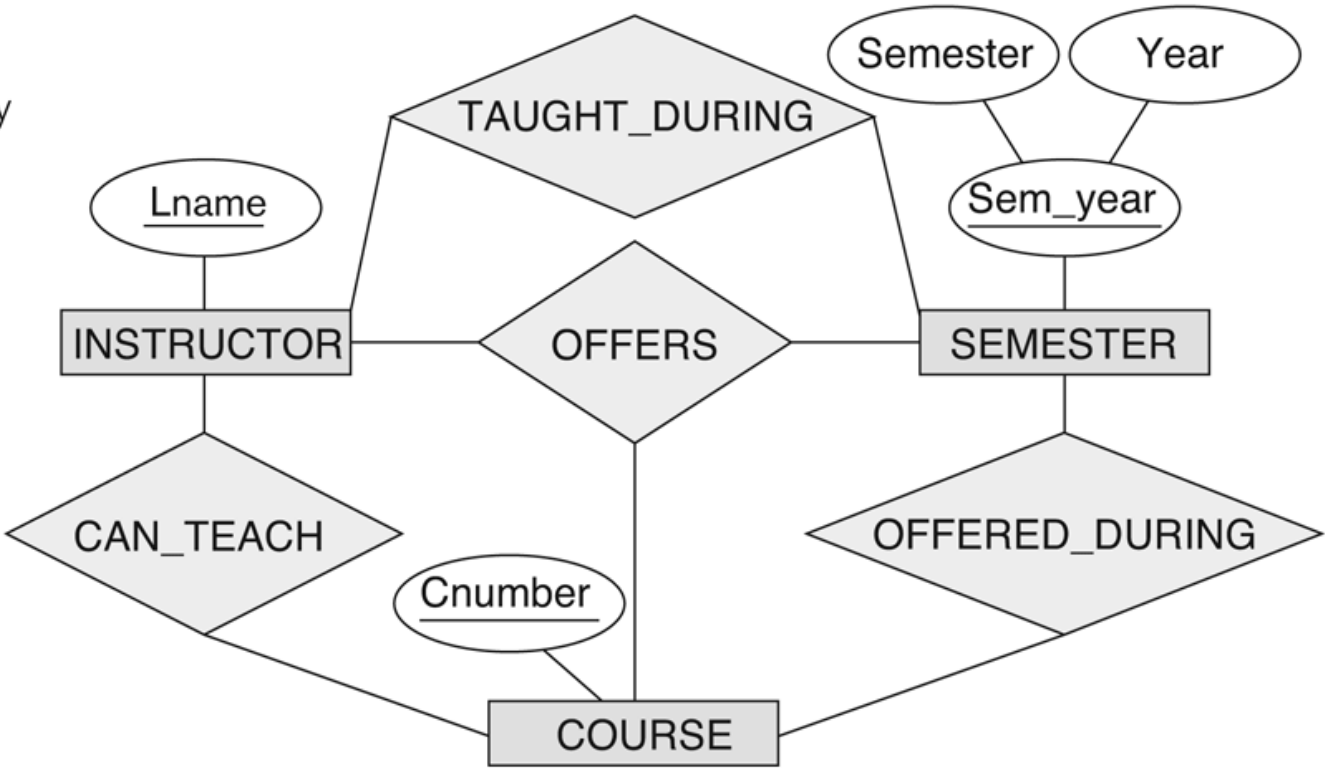
Discussion of n-ary relationships ($n > 2$)



- ✓ If a particular binary relationship can be derived from a higher-degree relationship at all times, then it is redundant
- ✓ For example, the TAUGHT_DURING binary relationship in Figure 3.18 (see next slide) can be derived from the ternary relationship OFFERS (based on the meaning of the relationships)

Another example of a ternary relationship

Figure 3.18
Another example of
ternary versus binary
relationship types.



Displaying constraints on higher-degree relationships



- ✓ The (min, max) constraints can be displayed on the edges – however, they do not fully describe the constraints
- ✓ Displaying a 1, M, or N indicates additional constraints
 - An M or N indicates no constraint
 - A 1 indicates that an entity can participate in at most one relationship instance *that has a particular combination of the other participating entities*
- ✓ In general, both (min, max) and 1, M, or N are needed to describe fully the constraints

Data Modeling Tools



- ✓ A number of popular tools that cover conceptual modeling and mapping into relational schema design.
 - Examples: ERWin, S- Designer (Enterprise Application Suite), ER- Studio, etc.
 - ✓ POSITIVES:
 - Serves as documentation of application requirements, easy user interface - mostly graphics editor support
 - ✓ NEGATIVES:
 - Most tools lack a proper distinct notation for relationships with relationship attributes
 - Mostly represent a relational design in a diagrammatic form rather than a conceptual ER-based design
- (See Chapter 12 for details)

Some of the Currently Available Automated Database Design Tools

COMPANY	TOOL	FUNCTIONALITY
Embarcadero Technologies	ER Studio	Database Modeling in ER and IDEF1X
	DB Artisan	Database administration, space and security management
Oracle	Developer 2000/Designer 2000	Database modeling, application development
Popkin Software	System Architect 2001	Data modeling, object modeling, process modeling, structured analysis/design
Platinum (Computer Associates)	Enterprise Modeling Suite: Erwin, BPWin, Paradigm Plus	Data, process, and business component modeling
Persistence Inc.	Pwertier	Mapping from O-O to relational model
Rational (IBM)	Rational Rose	UML Modeling & application generation in C++/JAVA
Resolution Ltd.	Xcase	Conceptual modeling up to code maintenance
Sybase	Enterprise Application Suite	Data modeling, business logic modeling
Visio	Visio Enterprise	Data modeling, design/reengineering Visual Basic/C++

Extended Entity-Relationship (EER) Model (in next chapter)

- ✓ The entity relationship model in its original form did not support the specialization and generalization abstractions
- ✓ Next chapter illustrates how the ER model can be extended with
 - Type-subtype and set-subset relationships
 - Specialization/Generalization Hierarchies
 - Notation to display them in EER diagrams

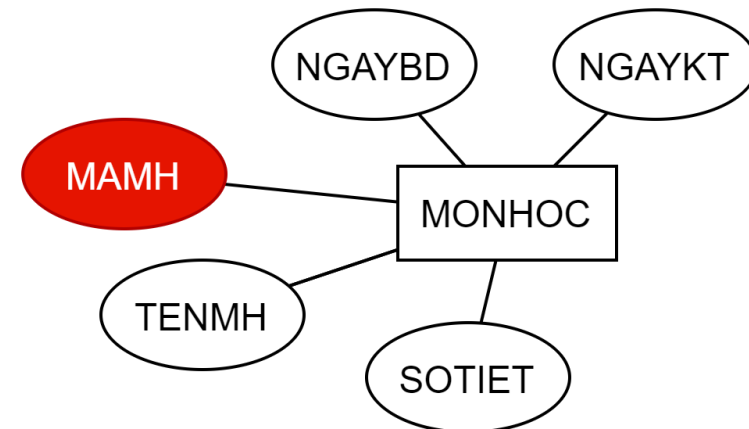
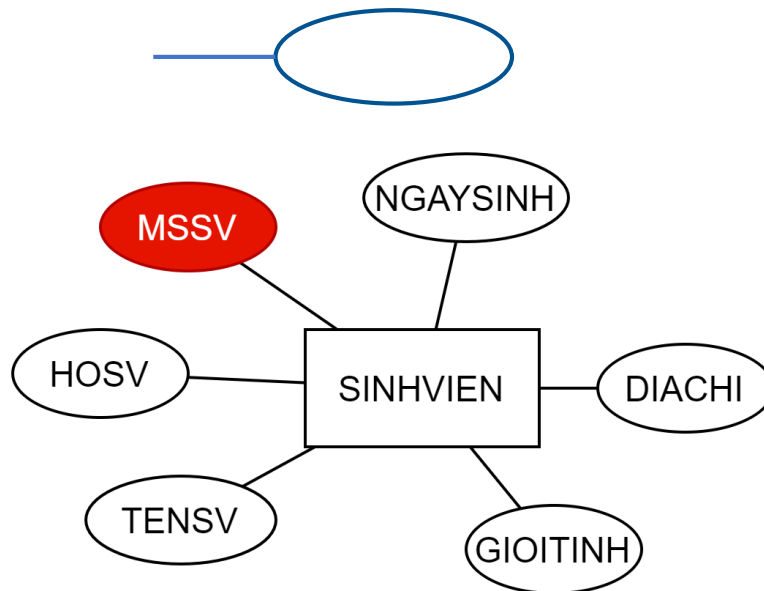
Chapter Summary

- ✓ ER Model Concepts: Entities, attributes, relationships
- ✓ Constraints in the ER model
- ✓ Using ER in step-by-step conceptual schema design for the COMPANY database
- ✓ ER Diagrams - Notation
- ✓ Alternative Notations – UML class diagrams, others

2.1.1 Thuộc tính

Thuộc tính (Attributes):

- Là những **đặc tính riêng biệt** của tập thực thể
- Tính chất của thực thể **cần được quản lý**
- Chỉ quan tâm đến các tính chất **có liên quan** đến ứng dụng
- Ký hiệu
- Ví dụ



2.1.2 Thuộc tính khóa

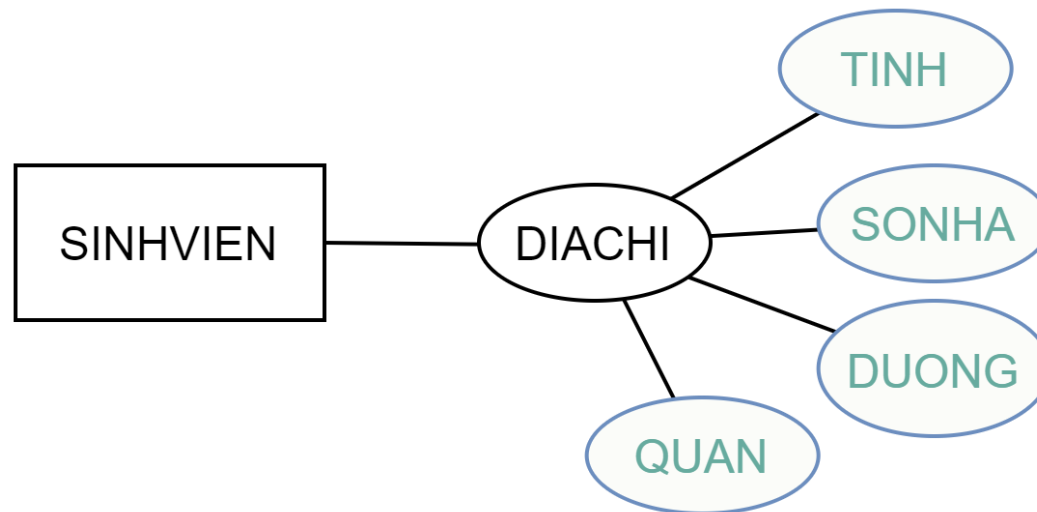
Thuộc tính khóa:

- ✓ Còn được gọi là thuộc tính định danh của tập thực thể (identifier)
- ✓ Dùng để **phân biệt** giữa **các thực thể** trong tập thực thể
- ✓ Khóa K của tập thực thể E là một hay nhiều thuộc tính sao cho
 - Lấy ra 2 thực thể bất kỳ e_1 , và e_2 trong E
 - Thì e_1 và e_2 không thể có các giá trị giống nhau tại các thuộc tính trong K
- ✓ Chú ý:
 - Mỗi tập thực thể phải có 1 khóa
 - Một khóa có thể **có 1** hay **nhiều** thuộc tính
 - Có thể có nhiều khóa trong 1 tập thực thể, ta sẽ chọn ra 1 khóa làm khóa chính cho tập thực thể đó
- ✓ Ví dụ: Mỗi sinh viên có một mã số duy nhất \Rightarrow Khóa của thực thể SINHVIEN là MSSV

2.1.3 Thuộc tính kết hợp

Thuộc tính kết hợp: là thuộc tính được kết hợp từ nhiều thuộc tính khác

- Ví dụ: thuộc tính **ĐỊA CHỈ** của tập thực thể **SINHVIÊN** có thể chia nhỏ thành các thành phần: **SỐ NHÀ**, **ĐƯỜNG**, **QUẬN**, **TỈNH**



2.1.4 Thuộc tính đa trị



Thuộc tính đa trị: là thuộc tính nhận nhiều giá trị đối với một thực thể cụ thể.

- Ví dụ: thuộc tính DIENTHOAI của tập thực thể SINHVIEN là thuộc tính đa trị. Mỗi giáo viên có thể có nhiều số điện thoại.

2.2 Mỗi kết hợp

Đặc điểm:

- ✓ Diễn tả mối quan hệ ngữ nghĩa giữa ít nhất 2 thực thể khác nhau
- ✓ Biểu diễn quan hệ kết hợp giữa 2 thực thể gọi là mỗi kết hợp nhị phân.
- ✓ Biểu diễn quan hệ kết hợp giữa nhiều thực thể (từ 3 trở lên) gọi là mỗi kết hợp đa phân.
- ✓ Có tên gọi riêng
- ✓ Có thuộc tính riêng của mỗi kết hợp
- ✓ Khóa hiển nhiên: không cần mô tả → được suy từ mô tả mỗi kết hợp (tổ hợp khóa)

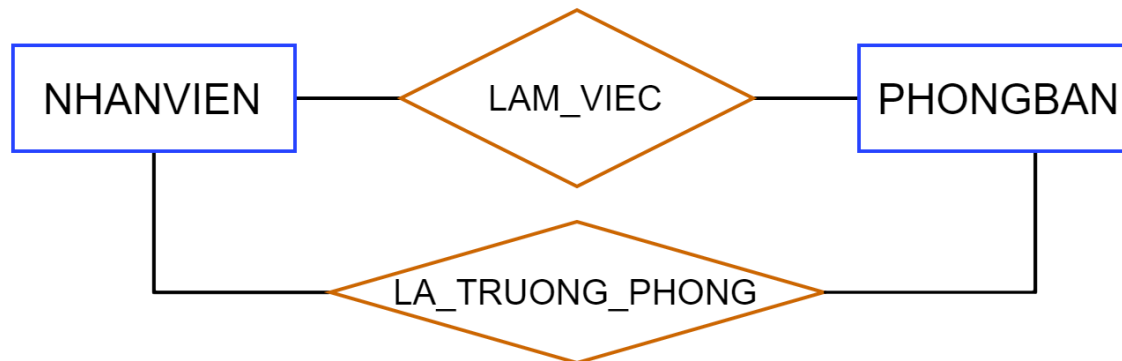
2.2 Mỗi kết hợp

✓ Ký hiệu:



✓ Ví dụ: giữa tập thực thể NHANVIEN và PHONGBAN có các liên kết

- Một nhân viên thuộc một phòng ban nào đó
- Một phòng ban có một nhân viên làm trưởng phòng



2.2.1 Bảng số

Xét mỗi kết hợp nhị phân R (binary relationship) giữa 2 tập thực thể E và F , tính multiplicity bao gồm:

✓ Một-Nhiều

- Một E có thể kết hợp với nhiều F
- Một F có thể kết hợp với một E



✓ Một-Một

- Một E có thể kết hợp với một F
- Một F có thể kết hợp với một E

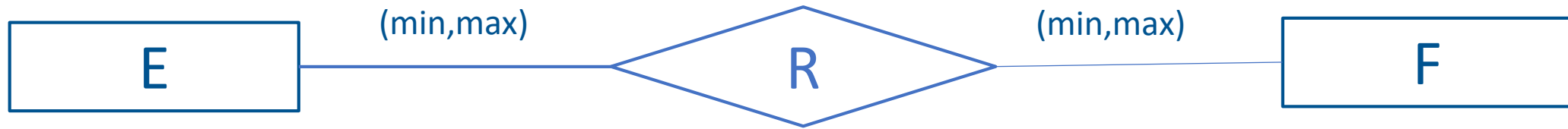


✓ Nhiều-Nhiều

- Một E có thể kết hợp với nhiều F
- Một F có thể kết hợp với nhiều E



2.2.1 Bảng số



✓ (min, max) chỉ định mỗi thực thể e thuộc tập các thực thể E tham gia **ít nhất** và **nhiều nhất** vào thể hiện của R

✓ Giải thích

- $(0,1)$: không hoặc một
- $(1,1)$: duy nhất một
- $(0,n)$: không hoặc nhiều
- $(1,n)$: một hoặc nhiều

2.2.1 Bảng số

✓ Một phòng ban có nhiều nhân viên



✓ Một nhân viên chỉ thuộc 1 phòng ban



✓ Một nhân viên có thể được phân công vào nhiều đề án hoặc không được phân công vào đề án nào



✓ Một nhân viên có thể là trưởng phòng của 1 phòng ban nào đó

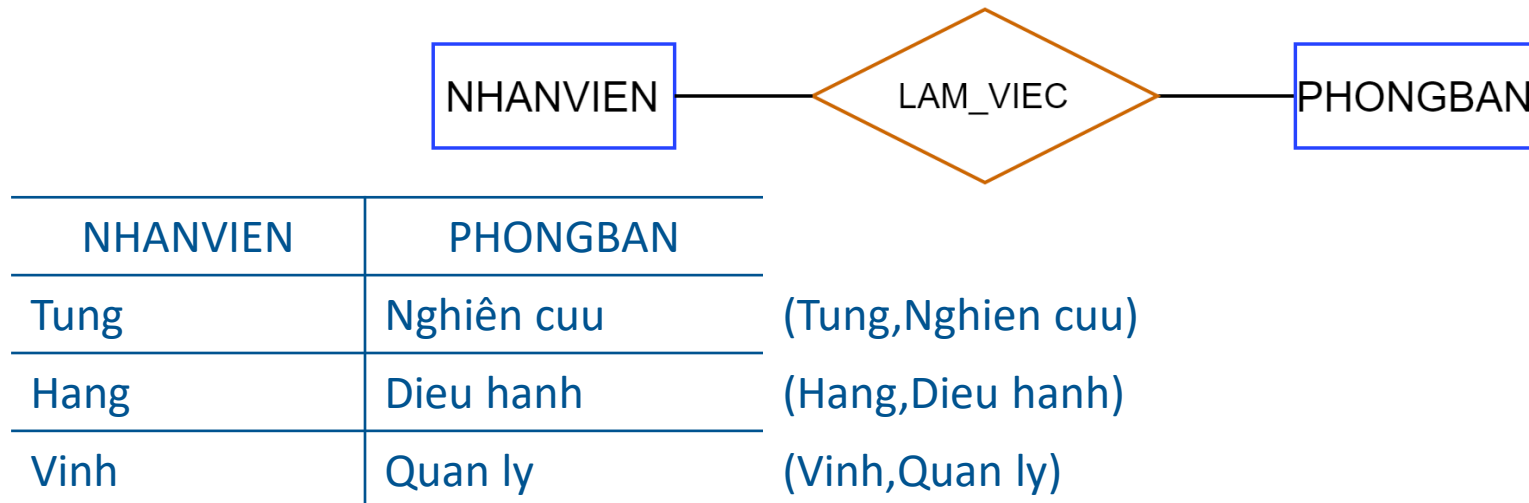


2.2.2 Thể hiện

Thể hiện CSDL còn chứa các mối kết hợp cụ thể

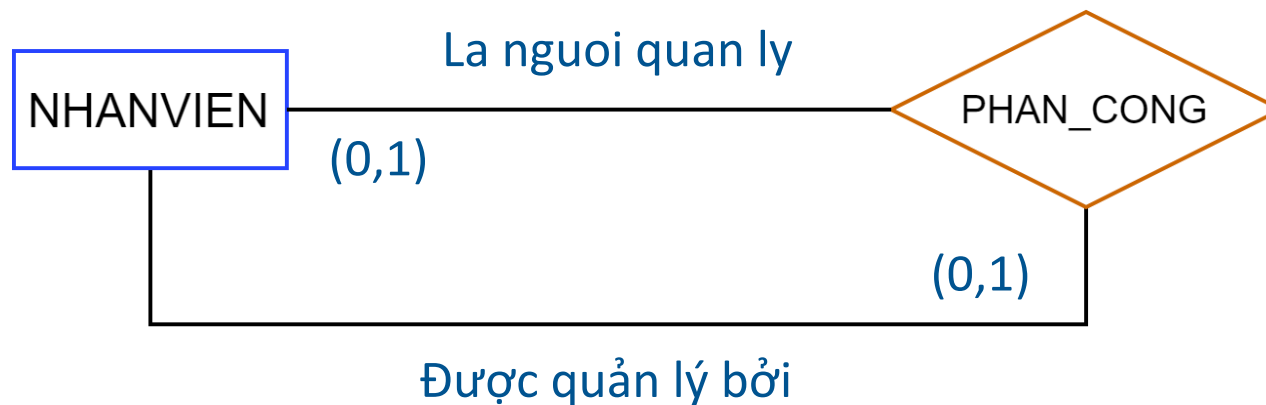
- Cho mỗi kết hợp R kết nối n tập thực thể E_1, E_2, \dots, E_n
- Thể hiện của R là tập hữu hạn các danh sách (e_1, e_2, \dots, e_n) , trong đó e_i là các giá trị được chọn từ các tập thực thể E_i .

Xét mối quan hệ:



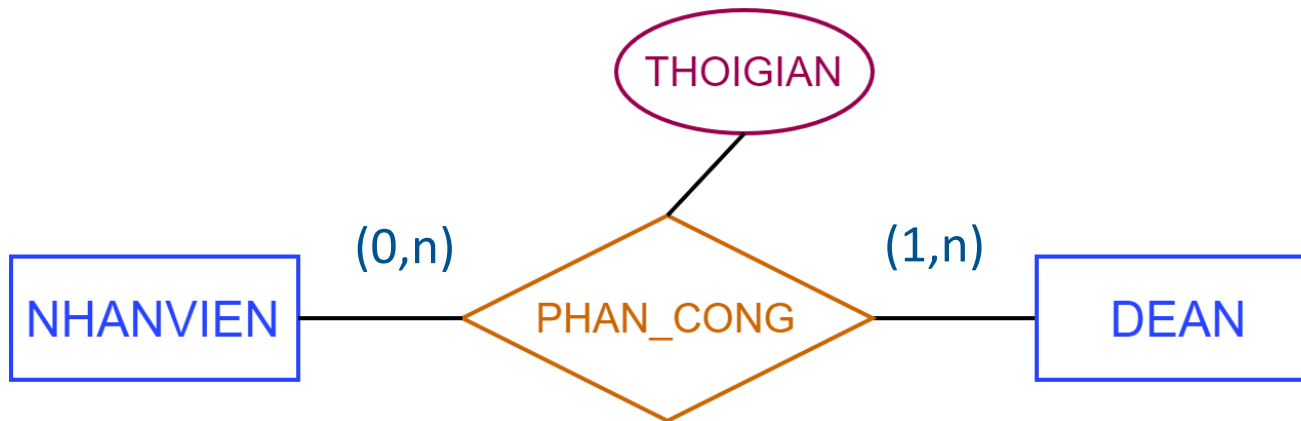
2.2 Mỗi kết hợp

Một loại thực thể có thể tham gia nhiều lần vào một quan hệ với nhiều vai trò khác nhau



2.2 Mỗi kết hợp

Thuộc tính trên mỗi quan hệ mô tả tính chất cho mỗi quan hệ đó và không thể gắn liền với những thực thể tham gia vào mỗi quan hệ



2.2.3 Ràng buộc trên mỗi kết hợp



- ✓ Nhằm giới hạn khả năng có thể kết hợp của các thực thể tham gia
- ✓ Xuất phát từ ràng buộc của thể giới thực
- ✓ Có hai loại ràng buộc mỗi kết hợp chính
 - Ràng buộc dựa trên bảng số
 - Ràng buộc dựa trên sự tham gia

2.2.3 Ràng buộc trên mỗi kết hợp

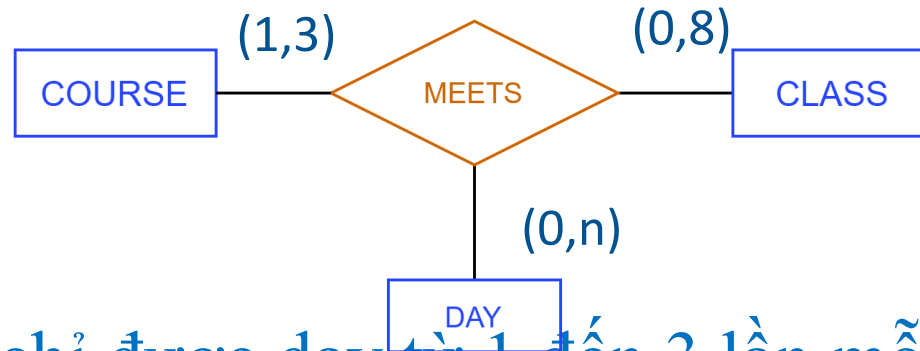
✓ Ví dụ:



- Sự tham gia của PEOPLE trong mỗi kết hợp là bắt buộc (mandatory participation), trong khi sự tham gia của CITY là tùy ý (optional participation)
- Diễn tả khái niệm một người sinh sống tại một thành phố duy nhất, trong khi thành phố có thể có nhiều người sinh sống

2.2.3 Ràng buộc trên mỗi kết hợp

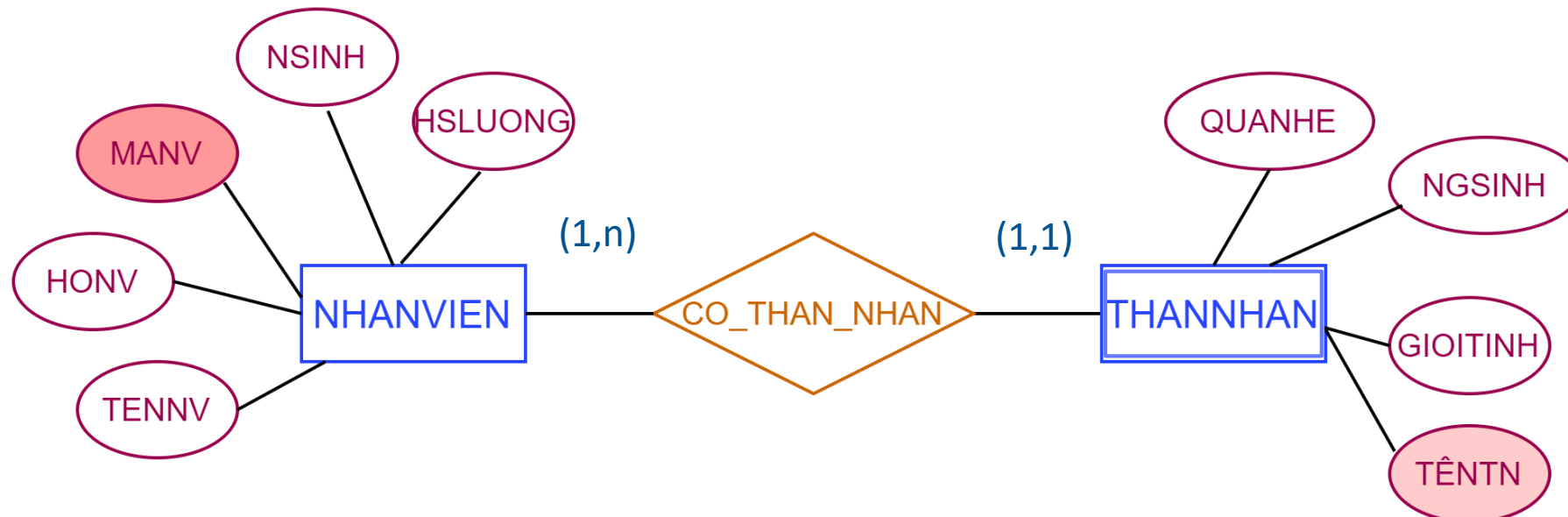
✓ Ví dụ:



- Mỗi môn học chỉ được dạy từ 1 đến 3 lần mỗi tuần, mỗi ngày trong tuần đều có một số buổi học nào đó, mỗi phòng học có tối đa 8 buổi học mỗi tuần
- Diễn tả khái niệm một môn học đã cho được dạy trong một ngày nào đó tại một phòng học nào đó

2.3 Thực thể yếu

- ✓ Là tập thực thể **có khóa** được tổ hợp từ những thuộc tính của tập thực thể khác
- ✓ Là loại thực thể không có thuộc tính khóa
- ✓ Tập thực thể yếu (weak entity set) phải tham gia vào mỗi kết hợp mà trong đó có một tập thực thể chính



3. Lược đồ ER (Entity-Relationship Diagram)



✓ Là đồ thị biểu diễn các tập thực thể, thuộc tính và mối quan hệ

✓ Định



Tập thực thể



Thuộc tính



Quan hệ

- ✓ Cạnh là đường nối giữa
- Tập thực thể và thuộc tính
 - Mối quan hệ và tập thực thể
 - Mối kết hợp và thuộc tính

3.1 Thể hiện của lược đồ E/R



Một CSDL được mô tả bởi lược đồ E/R sẽ chứa đựng những dữ liệu cụ thể gọi là thể hiện CSDL

- Mỗi tập thực thể sẽ có tập hợp hữu hạn các thực thể
 - Giả sử tập thực thể NHANVIEN có các thực thể như NV1, NV2, ...NVn
- Mỗi thực thể sẽ có 1 giá trị cụ thể tại mỗi thuộc tính
 - NV1 có TENNV=“Tung”, NGSINH=“08/12/1955”, PHAI=“Nam”
 - NV2 có TENNV= “Hang”, NGSINH=“07/19/1966”, PHAI=“Nu”

Chú ý:

- Không lưu trữ lược đồ E/R trong CSDL
 - Khái niệm trừu tượng
- Lược đồ E/R chỉ giúp ta thiết kế CSDL trước khi chuyển các quan hệ và dữ liệu xuống mức vật lý

3.2 Tiêu chuẩn chọn khái niệm



- ✓ **Chọn là thực thể** khi có thể xác định một số đặc trưng cơ bản như các thuộc tính, mối kết hợp, tổng quát hoá hay chuyên biệt hoá
- ✓ **Chọn là thuộc tính** khi đối tượng có cấu trúc nguyên tố đơn giản và không có các đặc trưng khác

3.3 Các bước thiết kế ERD

- (1) Xác định **tập thực thể**
- (2) Xác định **mối kết hợp**
- (3) Quyết định (**min, max**) cho mỗi kết hợp
- (4) Xác định **thuộc tính** và gán thuộc tính cho tập thực thể và mỗi kết hợp
- (5) Quyết định **miền giá trị** cho thuộc tính
- (6) Quyết định **thuộc tính khóa**

3.5 Một số công cụ vẽ mô hình

- MS Visio
- PowerDesigner
- DBMAIN
- <https://app.diagrams.net/>

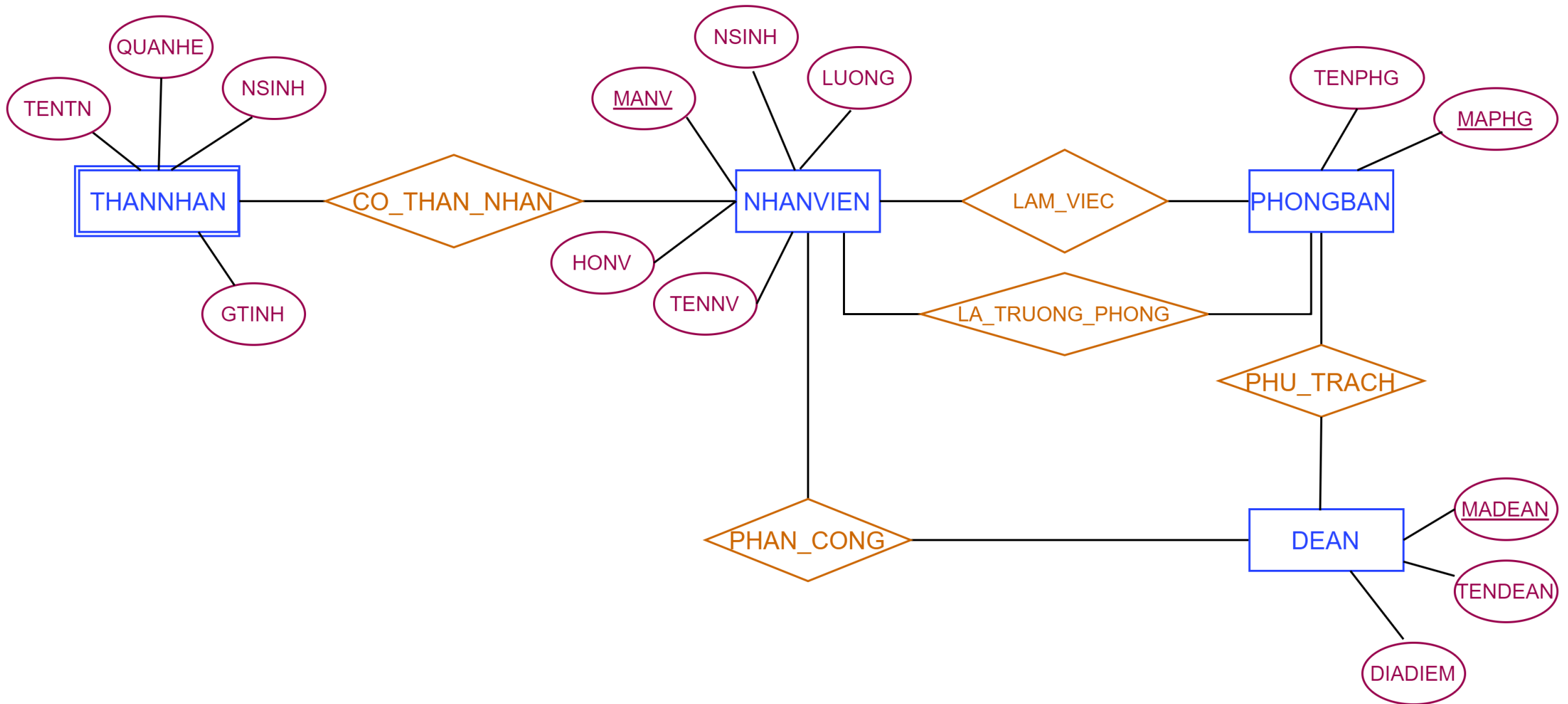
3.6 Chuyển ERD → Bảng

- ✓ Đây là bước chuyển đổi từ giai đoạn phân tích sang giai đoạn thiết kế
- ✓ Chuyển đổi ERD thành Mô hình vật lý \equiv BẢNG (lưu trữ và quản lý bởi DBMS trong các thiết bị lưu trữ)
- ✓ Quy tắc
 - Thực thể → Bảng
 - Mỗi kết hợp → bảng hoặc không là bảng tùy theo bản số
 - 1 – n : chuyển thành cha – con với ràng buộc khoá ngoại
 - m – n: chuyển thành thực thể
 - Ràng buộc khoá chính, ràng buộc khoá ngoại
 - Quy tắc vẽ các đường tham chiếu giữa các quan hệ

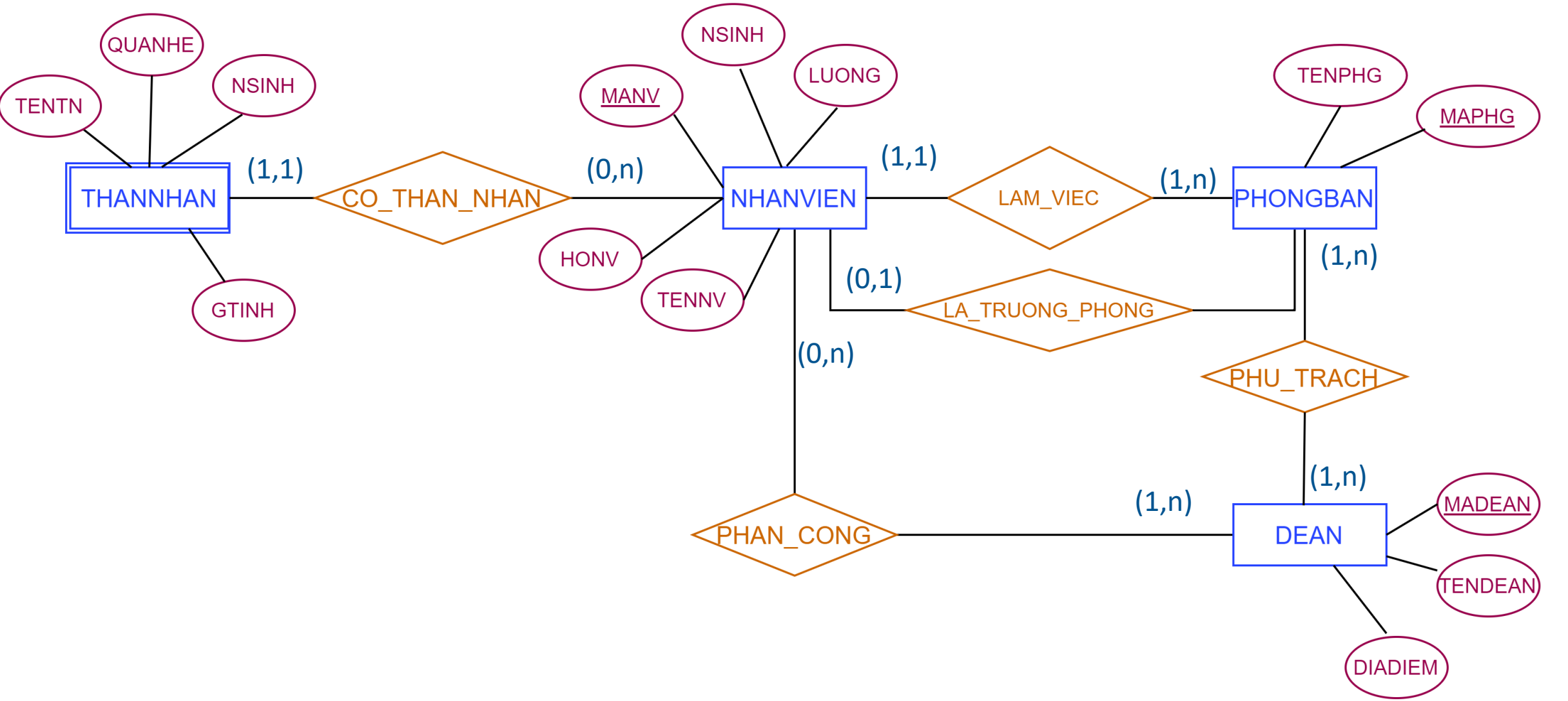
4. Ví dụ

- Theo dõi các thông tin liên quan tới nhân viên, phòng ban và đề án
- Công ty được tổ chức thành các phòng ban. Mỗi phòng ban có một tên duy nhất, mã số phòng ban duy nhất, và một nhân viên quản lý phòng đó. Ghi nhận lại ngày nhận chức trưởng phòng. Mỗi phòng ban có thể có nhiều địa điểm khác nhau
- Mỗi phòng ban chủ trì nhiều đề án, mỗi đề án có tên duy nhất, mã số đề án duy nhất và được triển khai ở một địa điểm
- Thông tin nhân viên của công ty được lưu trữ bao gồm mã nhân viên, địa chỉ, lương, phái, và ngày sinh. Mỗi nhân viên làm việc ở 1 phòng ban nhưng có thể tham gia nhiều đề án, trong đó đề án không nhất thiết phải do chính phòng ban của nhân viên chủ trì. Ghi nhận thông tin về thời gian tham gia đề án của nhân viên ứng với từng đề án tham gia, và cũng ghi nhận thông tin người quản lý trực tiếp nhân viên
- Mỗi nhân viên có thể có nhiều thân nhân, với mỗi thân nhân phải lưu trữ tên, phái, ngày sinh, và mối quan hệ với nhân viên trong công ty

4. Ví dụ



4. Ví dụ



4. Ví dụ

CSDL quản lý giáo vụ bao gồm các chức năng:

- Lưu trữ thông tin: Học viên, Giáo viên, Môn học
- Xếp lớp cho học viên, chọn lớp trưởng cho lớp
- Phân công giảng dạy: giáo viên dạy lớp nào, với môn học gì, ở học kỳ nào, năm học nào.
- Lưu trữ kết quả thi: học viên thi môn nào, lần thi thứ mấy, điểm thi bao nhiêu.



Hết

Xét các yêu cầu sau cho một cơ sở dữ liệu của một trường đại học dùng để quản lý quá trình học tập của các sinh viên:

- a. Mỗi sinh viên có tên, mã số sinh viên, số chứng minh nhân dân, địa chỉ tạm trú và số điện thoại, địa chỉ thường trú và số điện thoại, ngày sinh, giới tính, năm học (nhất, năm hai, năm ba, năm bốn, đã tốt nghiệp), ngành học (khoa), bậc học (cử nhân, thạc sĩ, tiến sĩ). Mỗi sinh viên có duy nhất một mã số và một số chứng minh nhân dân.
- b. Mỗi khoa được mô tả bởi tên, mã số khoa, địa chỉ, số điện thoại. Mỗi khoa có duy nhất một tên và một mã số.
- c. Mỗi môn học có tên, tóm tắt môn học, mã số, số tín chỉ, và được phụ trách bởi một khoa. Mỗi môn học có một mã số duy nhất.



d. Mỗi lớp học có một mã số, được tổ chức trong một học kỳ (học kỳ I, II) của một năm cụ thể, do một giảng viên phụ trách và dạy một môn học nào đó. Mã số của lớp học dùng để phân biệt các lớp dạy cùng một môn học trong cùng một thời điểm (học kỳ/năm), có giá trị tăng dần từ 1, 2, 3, ..., cho đến tổng số các lớp dạy môn học đó của mỗi học kỳ. Ví dụ trong học kỳ 1 năm 2004, lớp 1 học môn đồ họa máy tính do giảng viên Nguyễn Văn A phụ trách, lớp 2 cũng học môn đồ họa máy tính do giảng viên Trần Văn B phụ trách.

e. Bảng điểm của một sinh viên phải liệt kê chi tiết lớp học mà sinh viên đã từng tham dự và điểm số đạt được (0, 1, ..., 10).

Xây dựng sơ đồ thực thể - liên kết (ER). Xác định các thuộc tính khoá cho mỗi kiểu thực thể và các ràng buộc (lực lượng, tham gia) cho mỗi kiểu liên kết.