HOA SEN UNIVERSITY

Lecture 6
Entity Relationship (ER) Model

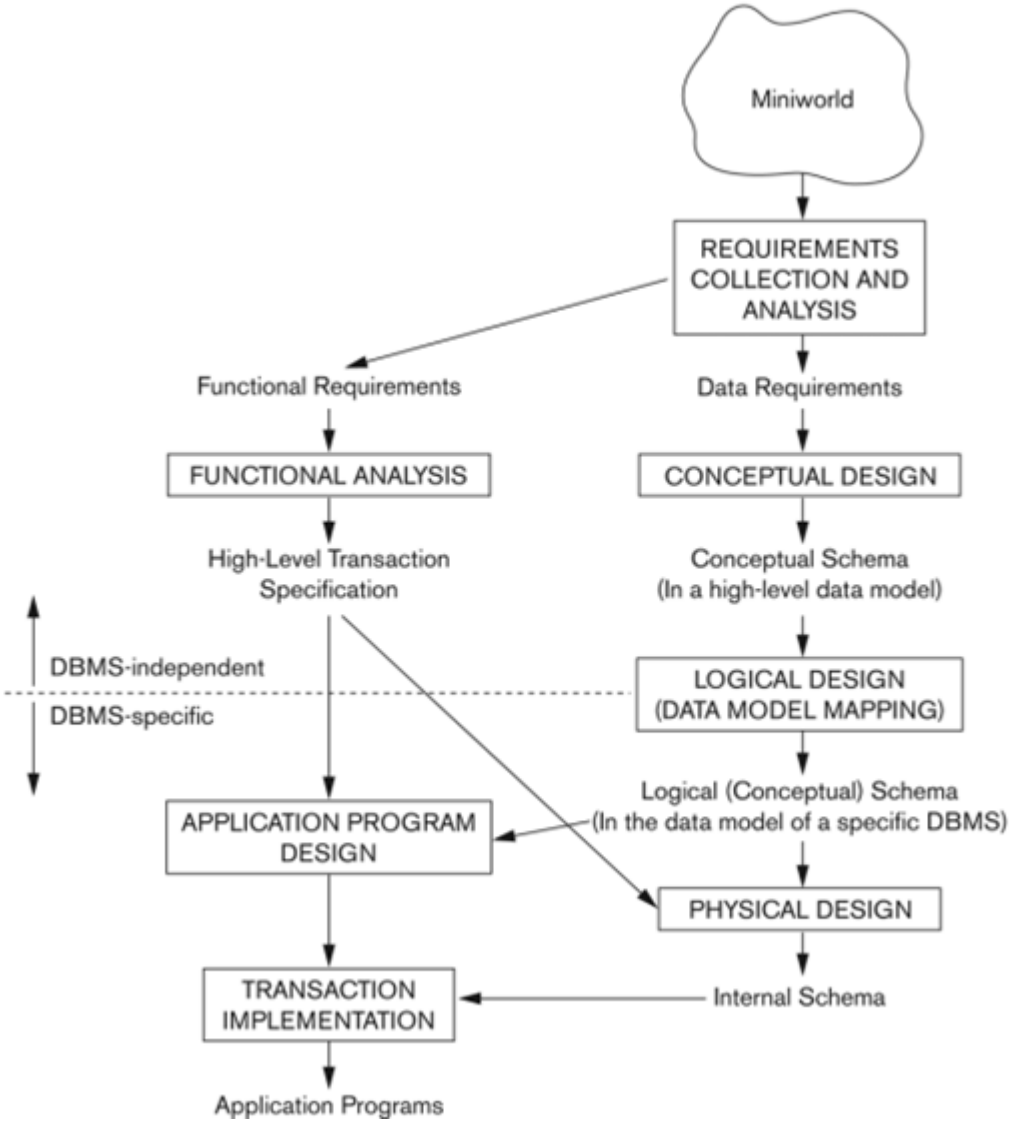
ĐẠI HỌC HOA SEN

Objectives

- Overview of Database Design Process
- ER Model Concepts
  - Entities and Attributes
  - Entity Types, Value Sets, and Key Attributes
  - Relationships and Relationship Types
  - Weak Entity Types
  - Roles and Attributes in Relationship Types
- ER Diagrams - Notation

- Ref.: Chapter 3

Overview of Database Design Process



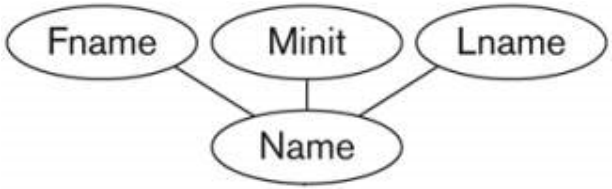
ER Model Concepts

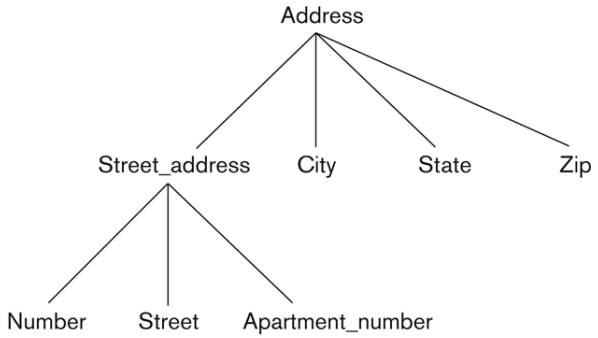
- Entities and Attributes
  - Entities are specific objects or things in the mini-world that are represented in the database.
    - For example the EMPLOYEE John Smith, the Research DEPARTMENT, the ProductX PROJECT
  - Attributes are properties used to describe an entity.
    - For example an EMPLOYEE entity may have the attributes FirstName, LastName, SSN, Address, Sex, BirthDate
  - A specific entity will have a value for each of its attributes.
    - For example a specific employee entity may have
      Name='John Smith', SSN='123456789',
      Address ='731, Fondren, Houston, TX',
      Sex='M', BirthDate='09-JAN-55‘
  - Each attribute has a *value set* (or data type) associated with it – e.g. integer, string, subrange, enumerated type, …

Types of Attributes

- Simple
  - Each entity has a single atomic value for the attribute.
    - For example: SSN or Sex attribute.

- Composite
  - The attribute may be composed of several components.
    - For example: Name(FirstName, MiddleName, LastName).
    - Composition may form a hierarchy where some components are themselves composite.

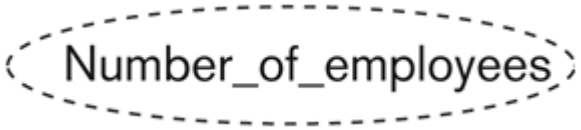




Types of Attributes (2)

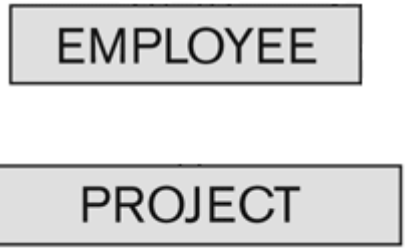
- Multi-valued
  - An entity may have multiple values for that attribute.
    - For example, Locations of a DEPARTMENT
    - Denoted as {Locations}

- Derived:
  - Attributes whose values are generated from other attributes using calculations, algorithms or procedures
    - Denoted as Number_of_employees

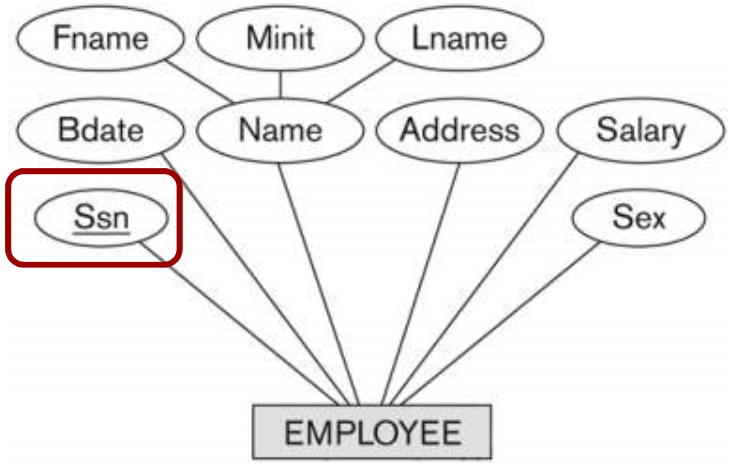


Entity Types and Key Attributes (1)

- **Entities** with the same basic attributes are grouped or typed into an entity type.
  - For example, the entity type EMPLOYEE and PROJECT.

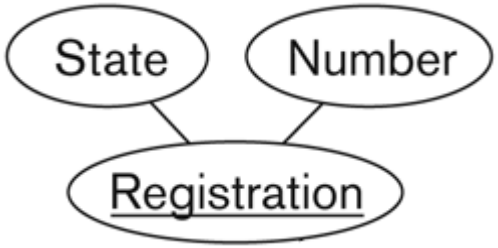


- Each **key** is <u>underlined</u>
- An attribute of an entity type for which each entity must have a unique value is called a key attribute of the entity type.
  - For example, SSN of EMPLOYEE.

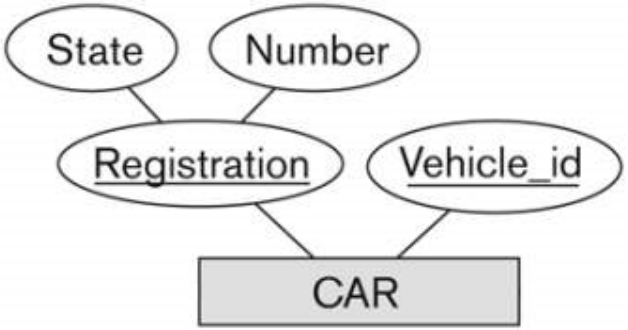


Entity Types and Key Attributes (2)

- 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.



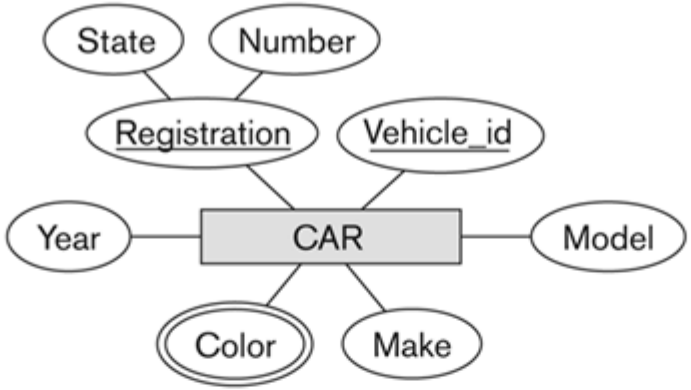
Entity Set

- **Entity set**: Each entity type will have a collection of entities stored in the database

- Entity set is the current *state* of the entities of that type that are stored in the database

Entity Type CAR with two keys and a corresponding Entity Set

(a)



(b)

CAR

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

CAR_1

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

CAR_2

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

CAR_3

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

ĐẠI HỌC HOA SEN

A Sample Database Application

- The COMPANY database keeps track of a company's employees, departments, and projects.
  - The company is organized into **departments**. Each department has *a unique name, a unique number, and a particular employee* who **manages** the department. We keep track of the start date when that employee began managing the department. A department may have several locations.
  - A department **controls** a number of **projects**, each of which has *a unique name, a unique number, and a single location*.
  - The database will store each **employee**'s *name, Social Security number, address, salary, sex (gender), and birth date*. An employee **is assigned** to one department, but may **work on** several projects, which are not necessarily controlled by the same department. It is required to keep track of the current number of hours per week that an employee works on each project, as well as the direct supervisor of each employee (who is another employee).
  - The database will keep track of the **dependents** of each employee for insurance purposes, **including** each dependent's *first name, sex, birth date, and relationship* to the employee.

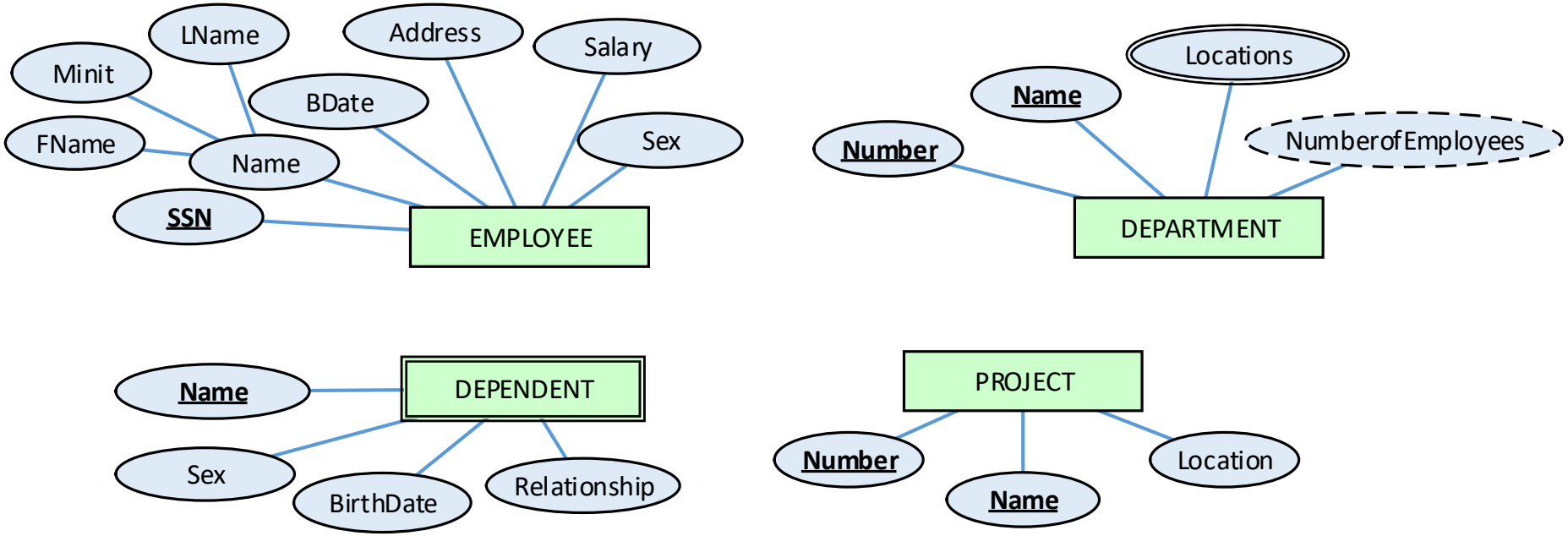
ĐẠI HỌC HOA SEN

Initial Design of Entity Types for the COMPANY Database Schema

- Based on the requirements, we can identify four initial entity types in the COMPANY database:
  - DEPARTMENT
  - PROJECT
  - EMPLOYEE
  - DEPENDENT [weak]
- The initial attributes shown are derived from the requirements description

Initial Design of Entity Types:

- EMPLOYEE, DEPARTMENT, PROJECT, DEPENDENT



ĐẠI HỌC HOA SEN

Refining the initial design by introducing relationships

- The initial design is typically not complete
- Some aspects in the requirements will be represented as **relationships**

- ER model has three main concepts:
  - Entities (and their entity types and entity sets)
  - Attributes (simple, composite, multivalued)
  - Relationships (and their relationship types and relationship sets)

Relationships and Relationship Types

- A **relationship** relates two or more distinct entities with a specific meaning.
  - For example,
    - EMPLOYEE John Smith *works on* the ProductX PROJECT,
    - 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,
    - 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.

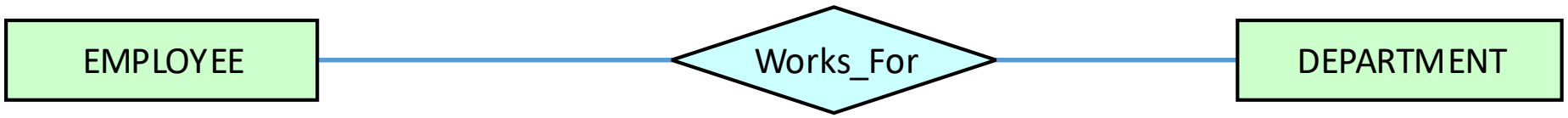
Relationship type vs. relationship set

- 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)

- 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

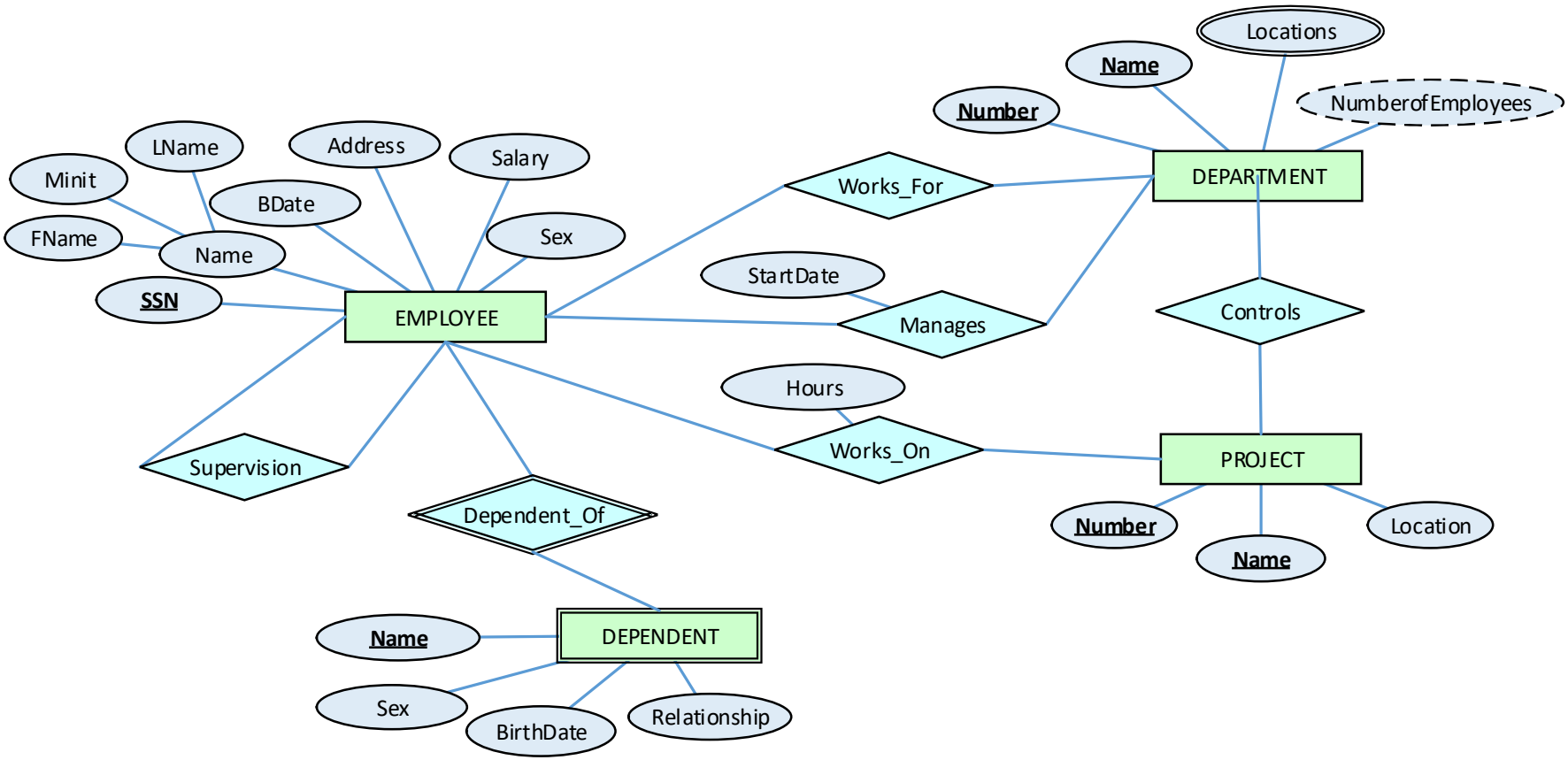


ĐẠI HỌC HOA SEN

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



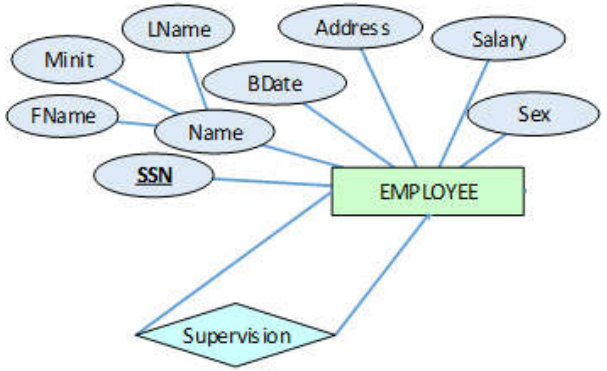
ĐẠI HỌC HOA SEN

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
  - ...
- 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.

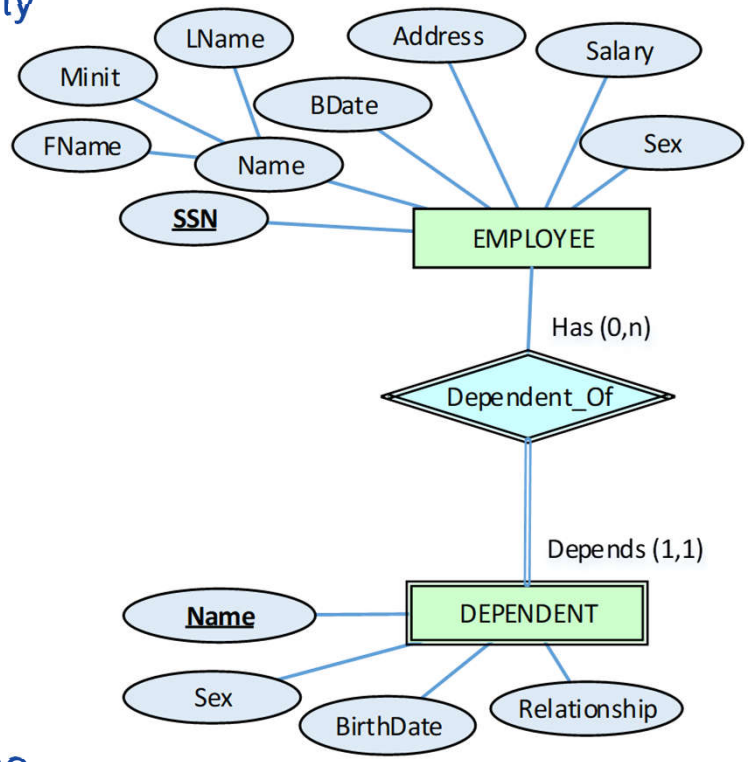
Recursive Relationship Type

- 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

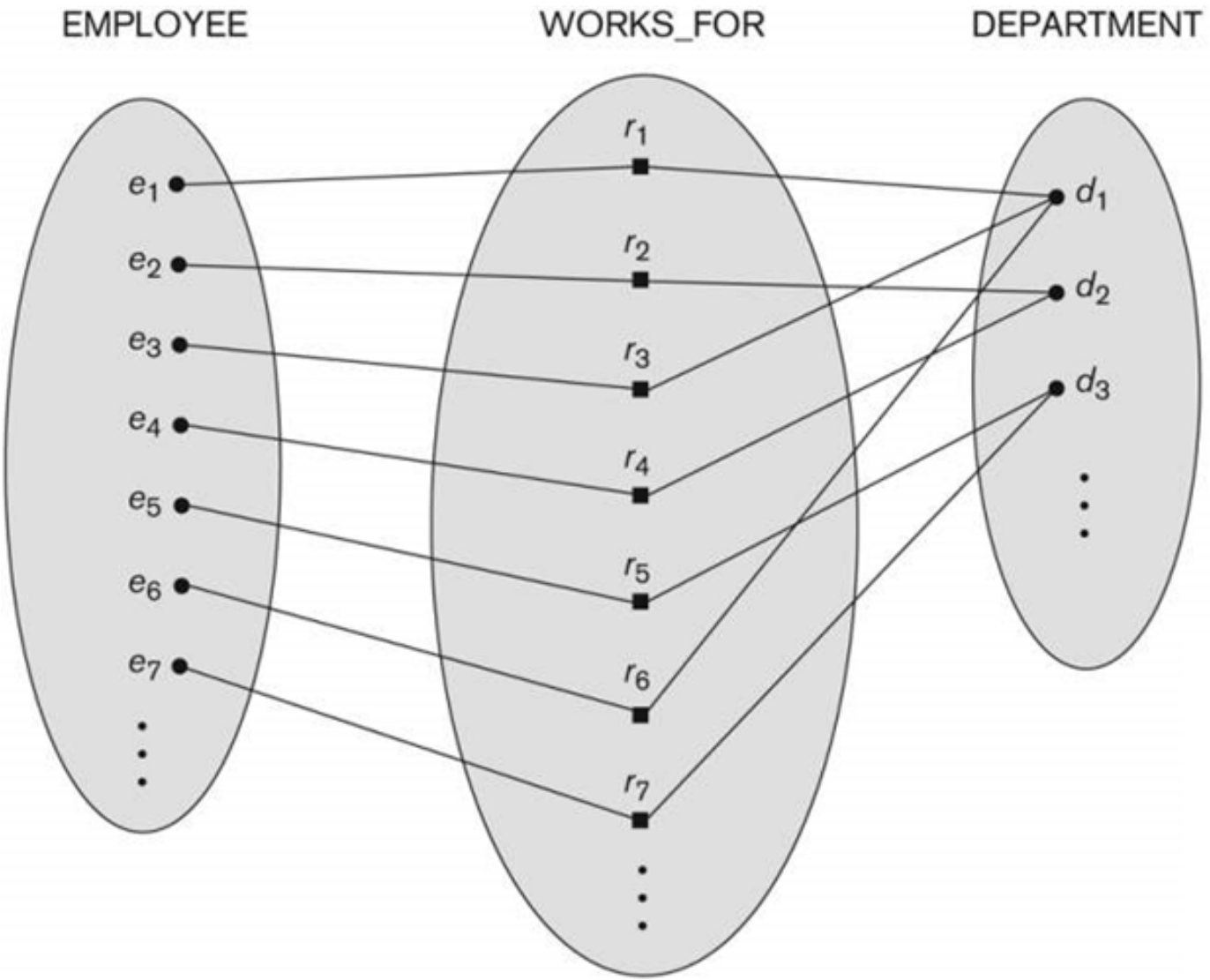


ĐẠI HỌC HOA SEN

Constraints on Relationships

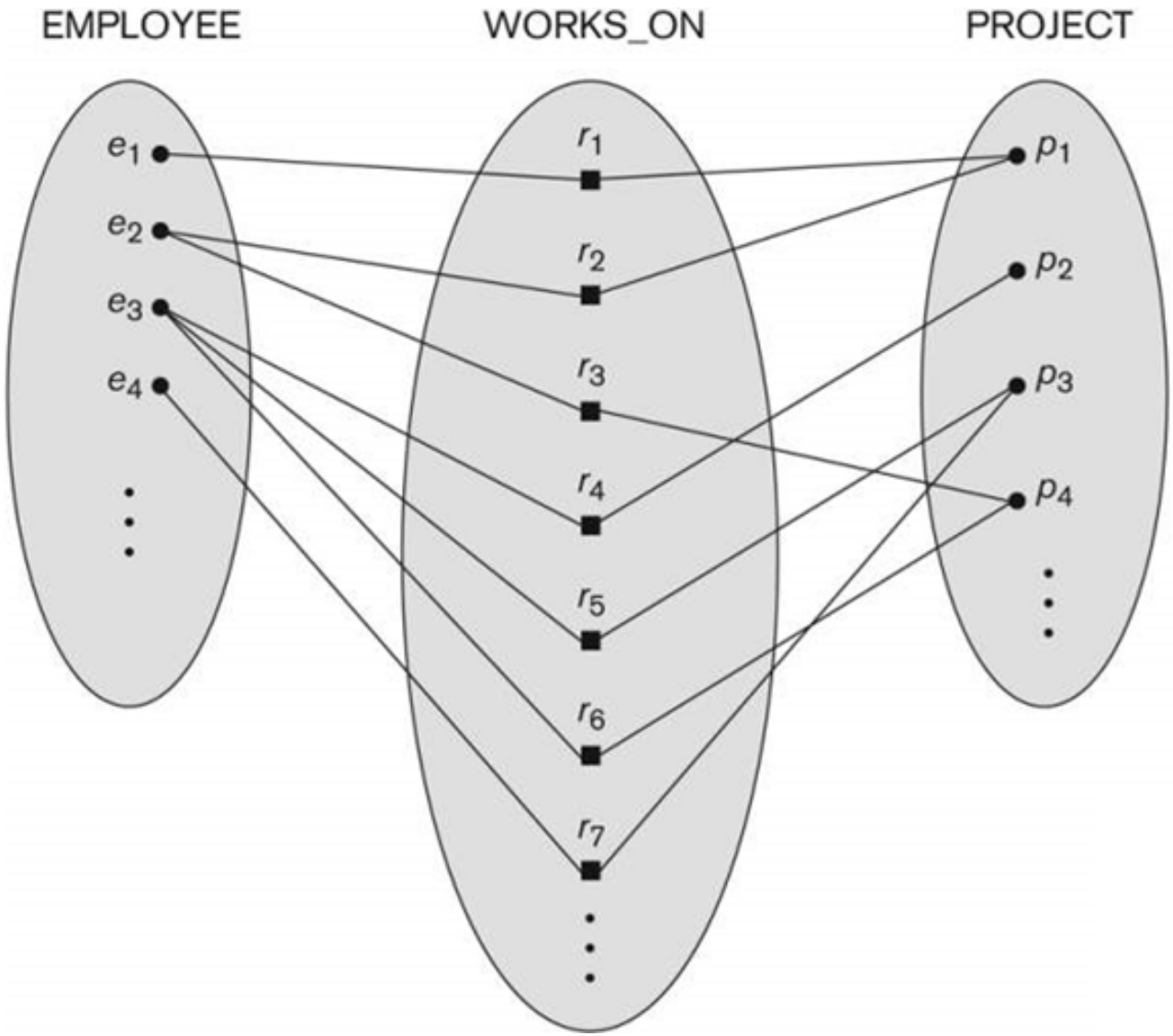
- Constraints on Relationship Types
  - (Also known as ratio constraints)
  - Cardinality Ratio (specifies *maximum* participation)
    - One-to-one (1:1)
    - One-to-many (1:N) or Many-to-one (N:1)
    - Many-to-many (M:N)
  - Existence Dependency Constraint (specifies *minimum* participation) (also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

Relationship instances of the WORKS_FOR N:1 relationship



ĐẠI HỌC HOA SEN

Relationship instances of the M:N WORKS_ON relationship



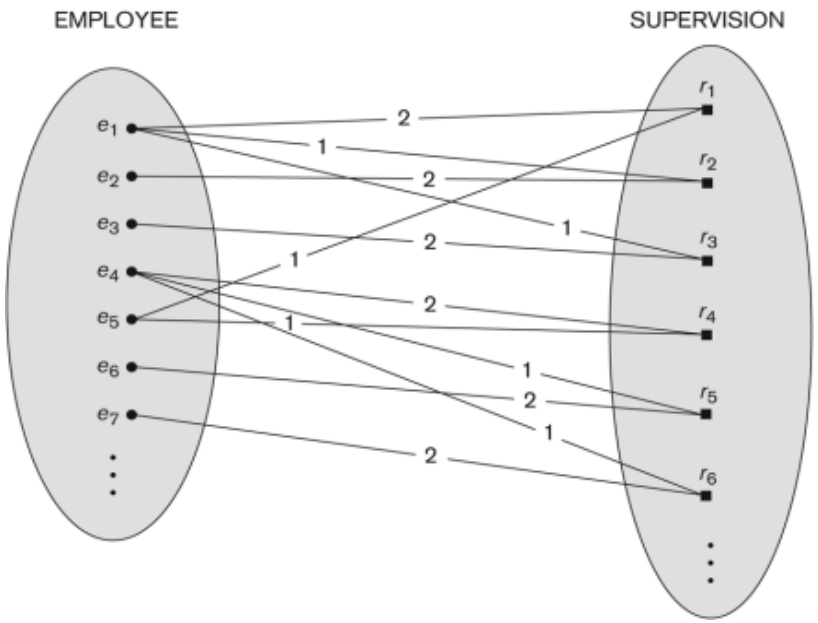
ĐẠI HỌC HOA SEN

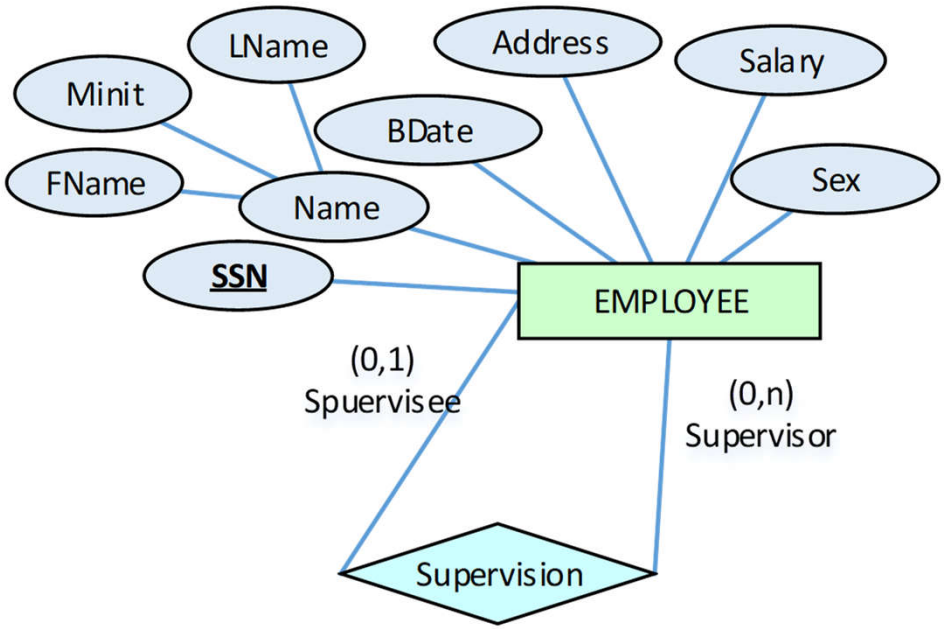
Displaying a recursive relationship

- 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 ER diagram, need to display role names to distinguish participations.

Recursive Relationship Type

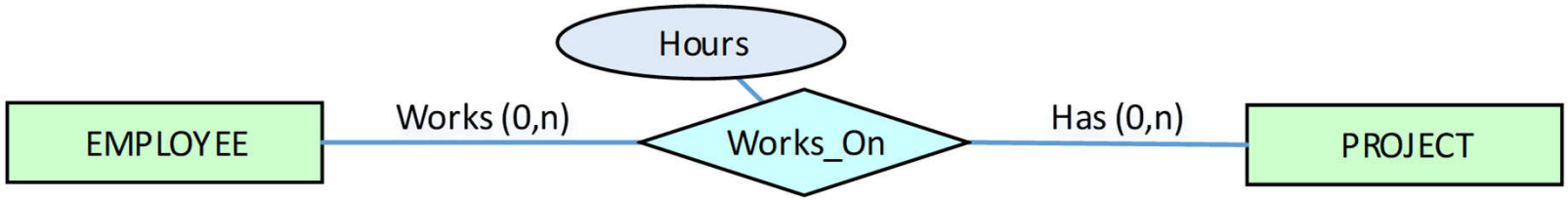
- SUPERVISION
  (participation role names are shown)





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

ĐẠI HỌC HOA SEN

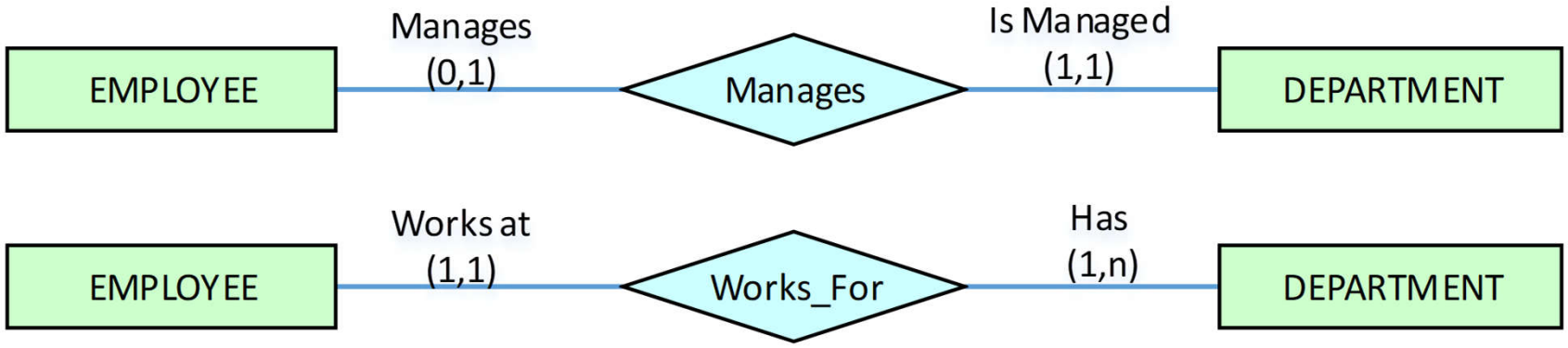
Notation for Constraints on Relationships

- 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.

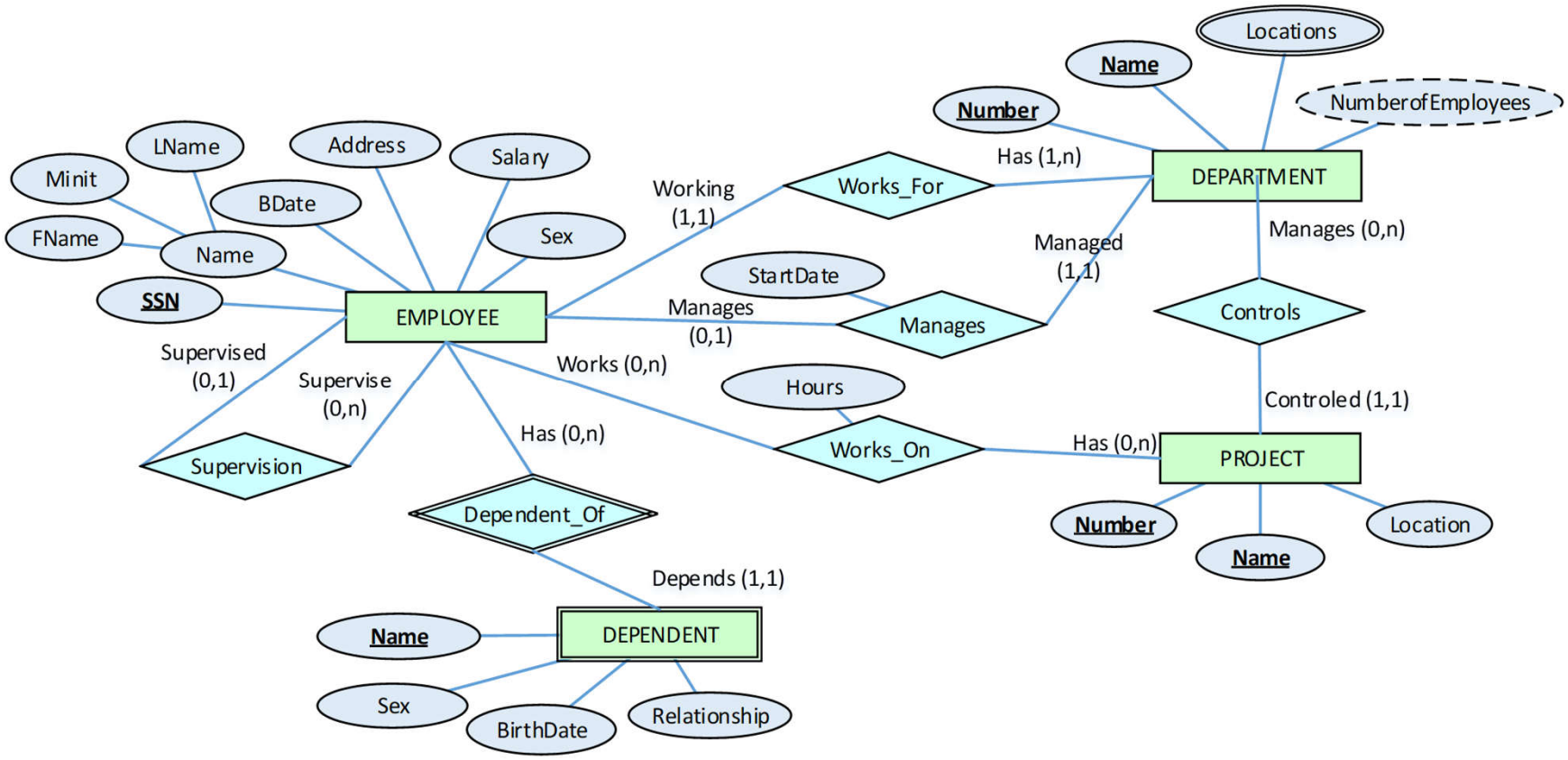
Alternative (min, max) notation for relationship structural constraints:

- 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 min ≤ max, min ≥ 0, max ≥ 1
- Derived from the knowledge of mini-world constraints

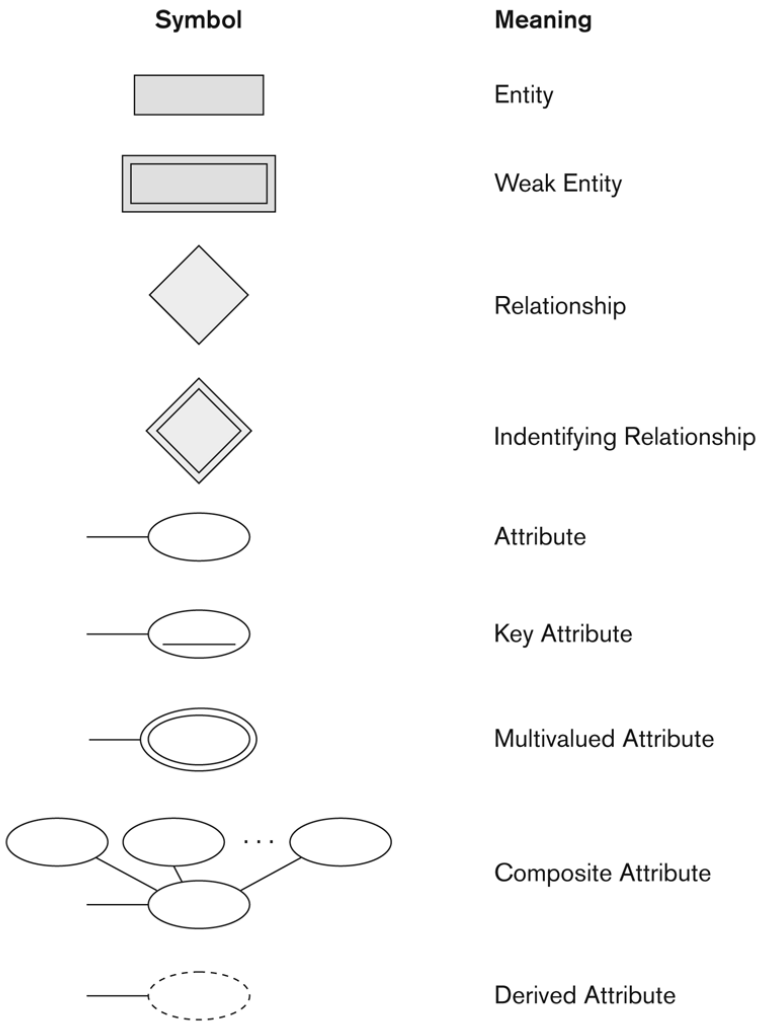


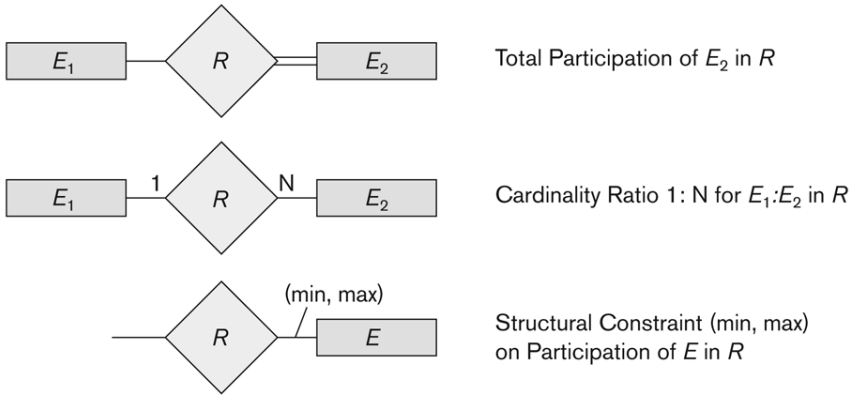
Read the min,max numbers next to the entity type and looking **away from** the entity type

COMPANY ER Schema Diagram using (min, max) notation



Notation for ER diagrams





ĐẠI HỌC HOA SEN

Relationships of Higher Degree

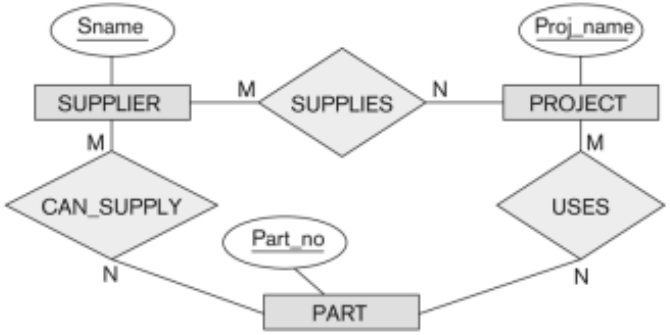
- Relationship types of degree 2 are called **binary**
- Relationship types of degree 3 are called **ternary** and of degree n are called **n-ary**
- In general, an n-ary relationship is not equivalent to n binary relationships
- Constraints are harder to specify for higher-degree relationships (n > 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 a)
- If needed, the binary and n-ary relationships can all be included in the schema design (see figure b)

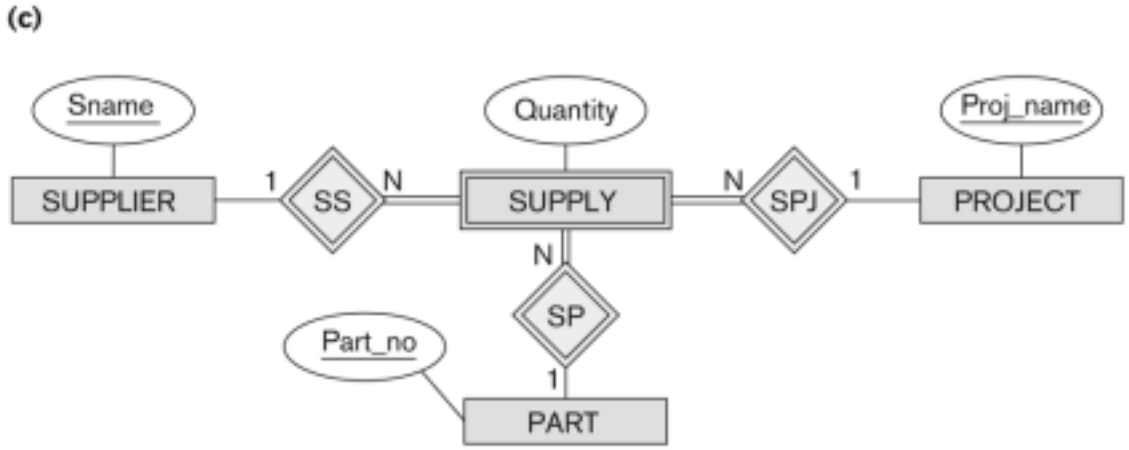
(a)

(b)



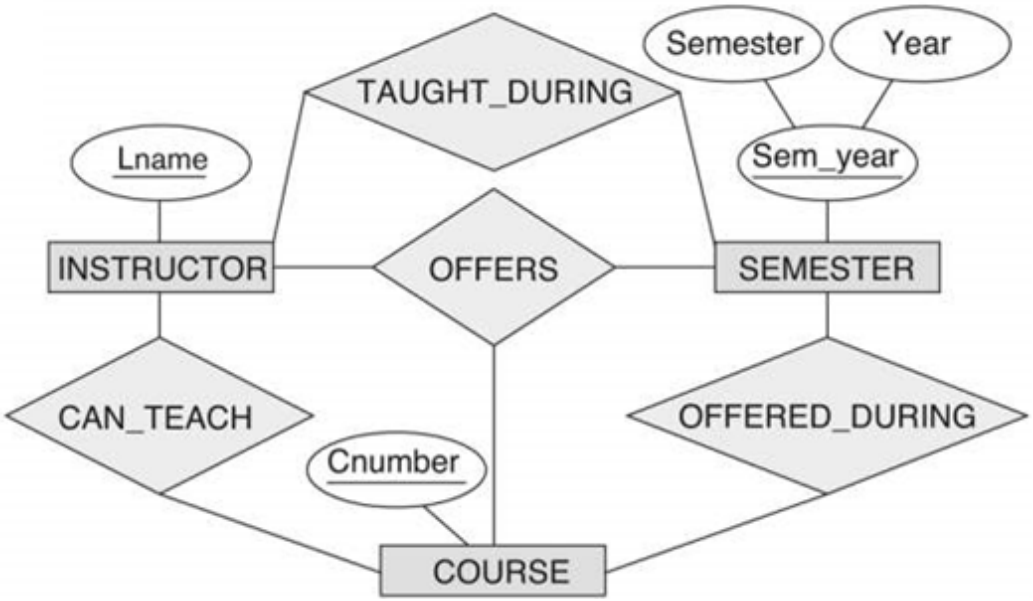
Discussion of n-ary relationships (n > 2)

- 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 c)



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 can be derived from the ternary relationship OFFERS (based on the meaning of the relationships)



ĐẠI HỌC HOA SEN

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

ĐẠI HỌC
HOA SEN

Q & A