

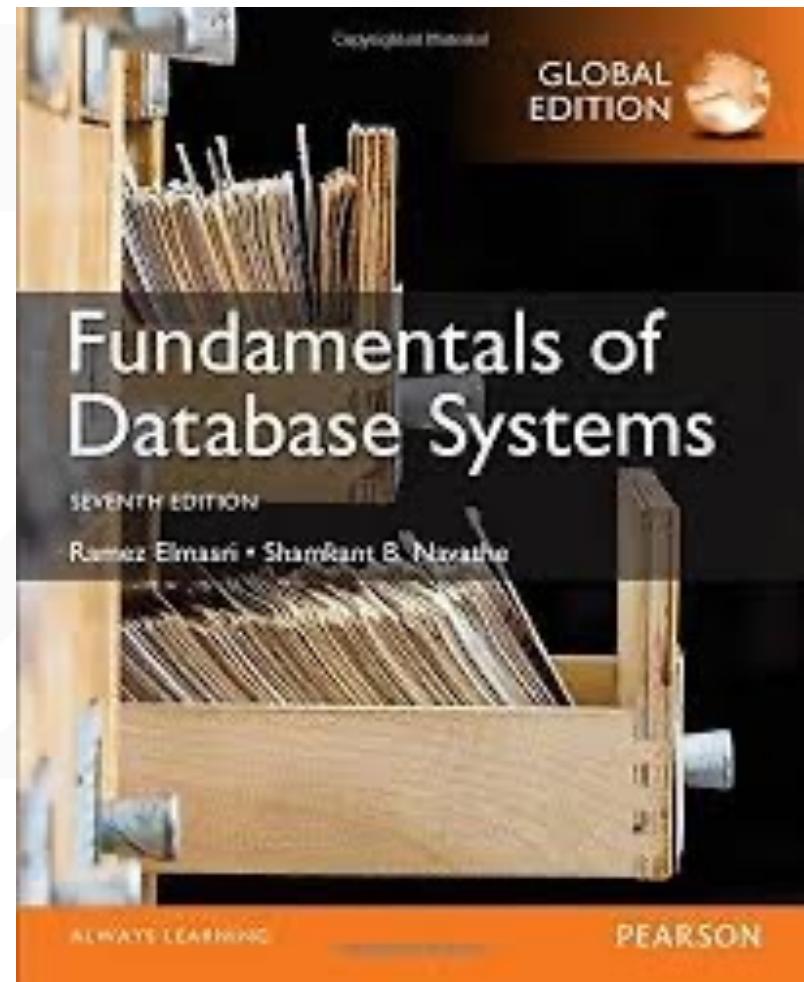
# Database Systems

Program in Computer Engineering  
School of Engineering

King Mongkut's Institute of Technology Ladkrabang

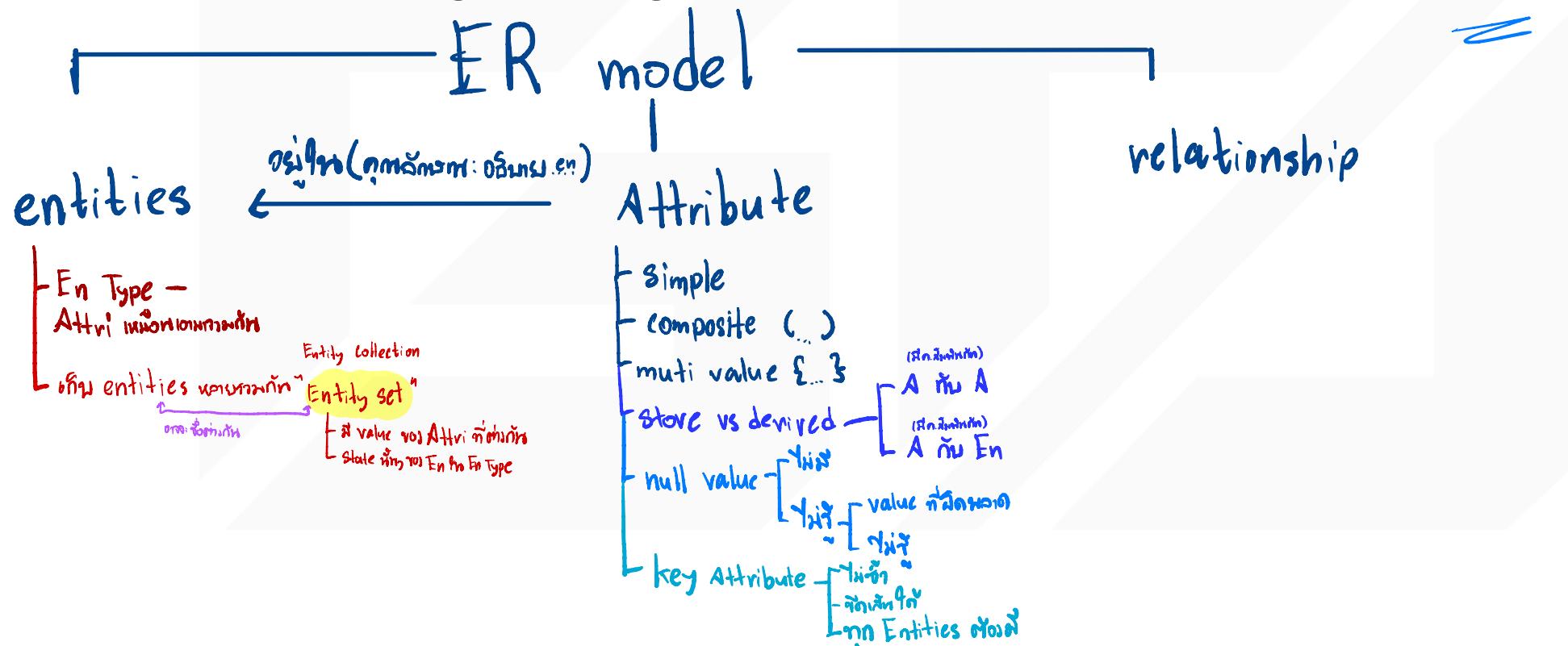
# Text

- Ramez Elmasri and Shamkant B. Navathe.  
“Fundamentals of Database Systems”  
7<sup>th</sup> Edition., Pearson, 2017



# Chapter 3

## Data Modeling Using the Entity-Relationship (ER) Model



# Data Models (from Ch 2)

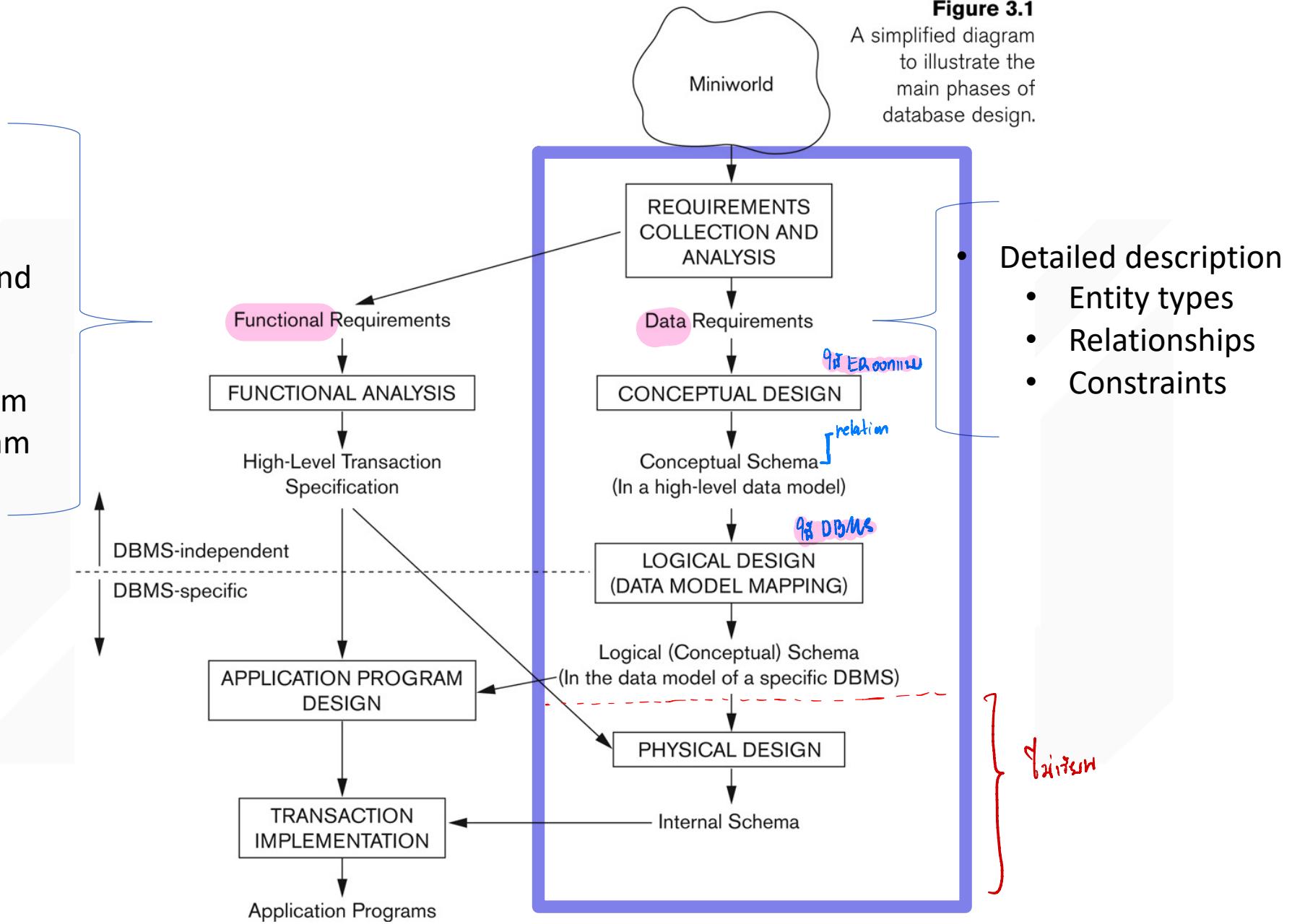
- A set of concepts to describe the ***structure*** of a database, the ***operations*** for manipulating these structures, and certain ***constraints*** that the database should obey.

# Overview of Database Design Process

- Two main activities:
  - Database design
  - Applications design
- Focus in this chapter on conceptual database design
  - To design the conceptual schema for a database application
- Applications design focuses on the programs and interfaces that access the database
  - Generally considered part of software engineering

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

- User defined operations (or transactions) including retrievals and updates.
- Common tools
  - Data flow diagram
  - Sequence diagram
  - Scenarios etc.



# Example COMPANY Database

- Create a database schema design based on the following (simplified) **requirements** of the COMPANY Database:

- The company is organized into DEPARTMENTS.  
Each department has a name, number and an employee who manages the department.

เจ้าหน้าที่ฝ่ายงาน: → ผู้ดูแล

We keep track of the start date of the department manager.

A department may have several locations.

สถานที่

- Each department controls a number of PROJECTS.  
Each project has a unique name, unique number and is located at a single location.

โครงการ

หมายเลข ประจำ project

- The database will store each **EMPLOYEE**'s social security number, address, salary, sex, and birthdate.
  - Each employee *works for one department* but may *work on several projects*.
  - The DB will keep *track* of the *number of hours per week* that an employee currently works on *each project*.
  - It is required to keep track of the *direct supervisor* of each employee

บุตร (ສໍານັກຝ່າດ)

ກາງຕົກມົນຕົມ employee ອີເບີບລົວດິກາ

- Each employee may have a number of DEPENDENTS.
  - For each dependent, the DB keeps a record of name, sex, birthdate, and relationship to the employee

ການສໍານັກຝ່າດ employee ໂພນ ຖະ ດູກສະກາ

# Entities and Attributes

- Entity is a basic concept for the ER model.

Entities are specific things or objects in the mini-world that are represented in the database.

- For example the EMPLOYEE John Smith, the Research DEPARTMENT, the ProductX PROJECT

คือตัวอย่าง entity

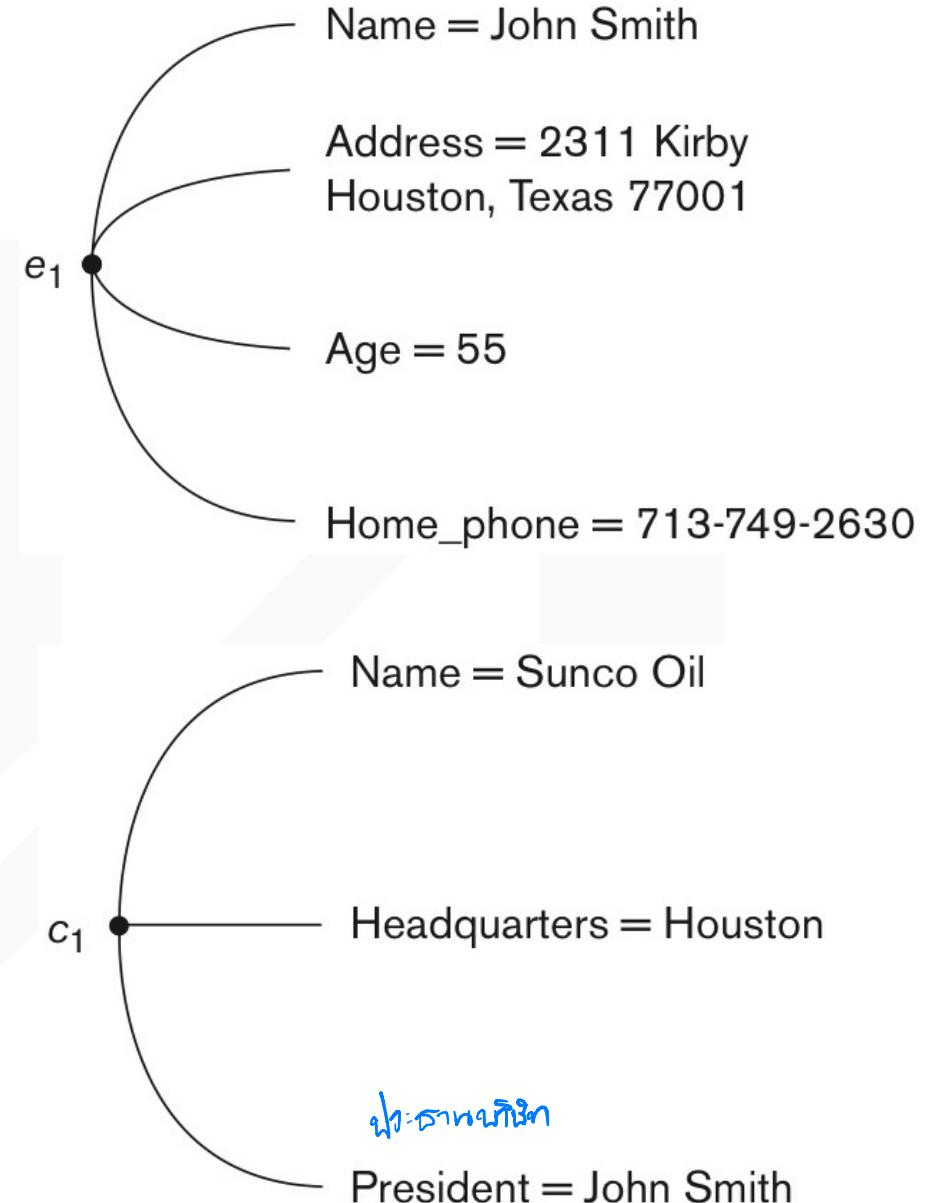
- Attributes are properties used to describe an entity.

- For example an EMPLOYEE entity may have the attributes Name, SSN, Address, Sex, BirthDate

\* សិក្សាក្នុងការអនុវត្តន៍របស់ខ្លួន និងរូបរាងរបស់ខ្លួន

# Entities and Attributes

- 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, date, enumerated type, ...

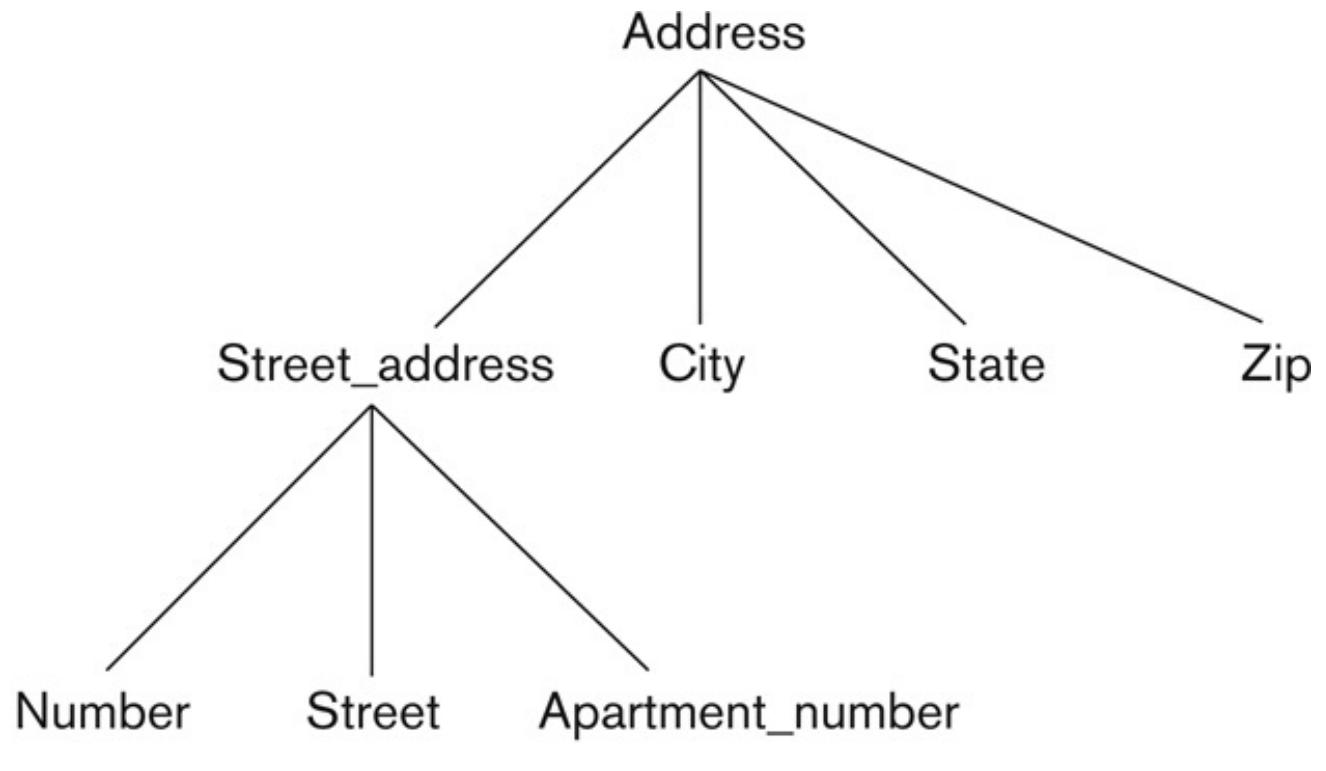


# Types of Attributes

- **Simple**   
  - Each entity has a single atomic value for the attribute.
  - For example, SSN or Sex.
- **Composite**   
  - The attribute may be composed of several components.
  - For example:
    - Address(Apt#, House#, Street, City, State, ZipCode, Country), or
    - Name(FirstName, MiddleName, LastName).
    - Composition may form a hierarchy where some components are themselves composite.
- **Multi-valued**   
  - An entity may have multiple values for that attribute.
  - For example,
    - Color of a CAR or
    - PreviousDegrees of a STUDENT.
  - Denoted as {Color} or {PreviousDegrees}.

ພົນກົດລົງ

- In general, composite and multi-valued attributes may be nested arbitrarily to any number of levels, although this is rare.
  - For example,
    - PreviousDegrees of a STUDENT is a composite multi-valued attribute denoted by {PreviousDegrees (College, Year, Degree, Field)}
    - Multiple PreviousDegrees values can exist
    - Each has four subcomponent attributes:
      - College, Year, Degree, Field



**Figure 3.4**  
A hierarchy of composite attributes.

↳ Attributed ត្រូវបានកំណត់ឡើង

រាយការណ៍នា Attributed ចិត្តថាគ្នុង

### • Stored vs. Derived Attributed ត្រូវបានគេដាក់ឡើង

- Two or more attribute values are related

• E.g., Birth\_date vs. Age

ចាប់ពីការណក ទំនាក់ទំនង

• The Age attribute is derived from Birth\_date.

• Age is called a "derived attribute" and is derivable from the Birth\_date.

- Some attribute values can be derived from related entities

• E.g., Number\_of\_employees of a DEPARTMENT entity

• Can be derived by counting the number of employee related to (working for) that department.

ព័ត៌មានទូទៅ គ្នា Null នៃយើង  
ដោយពេលបែក ការណែនាំ

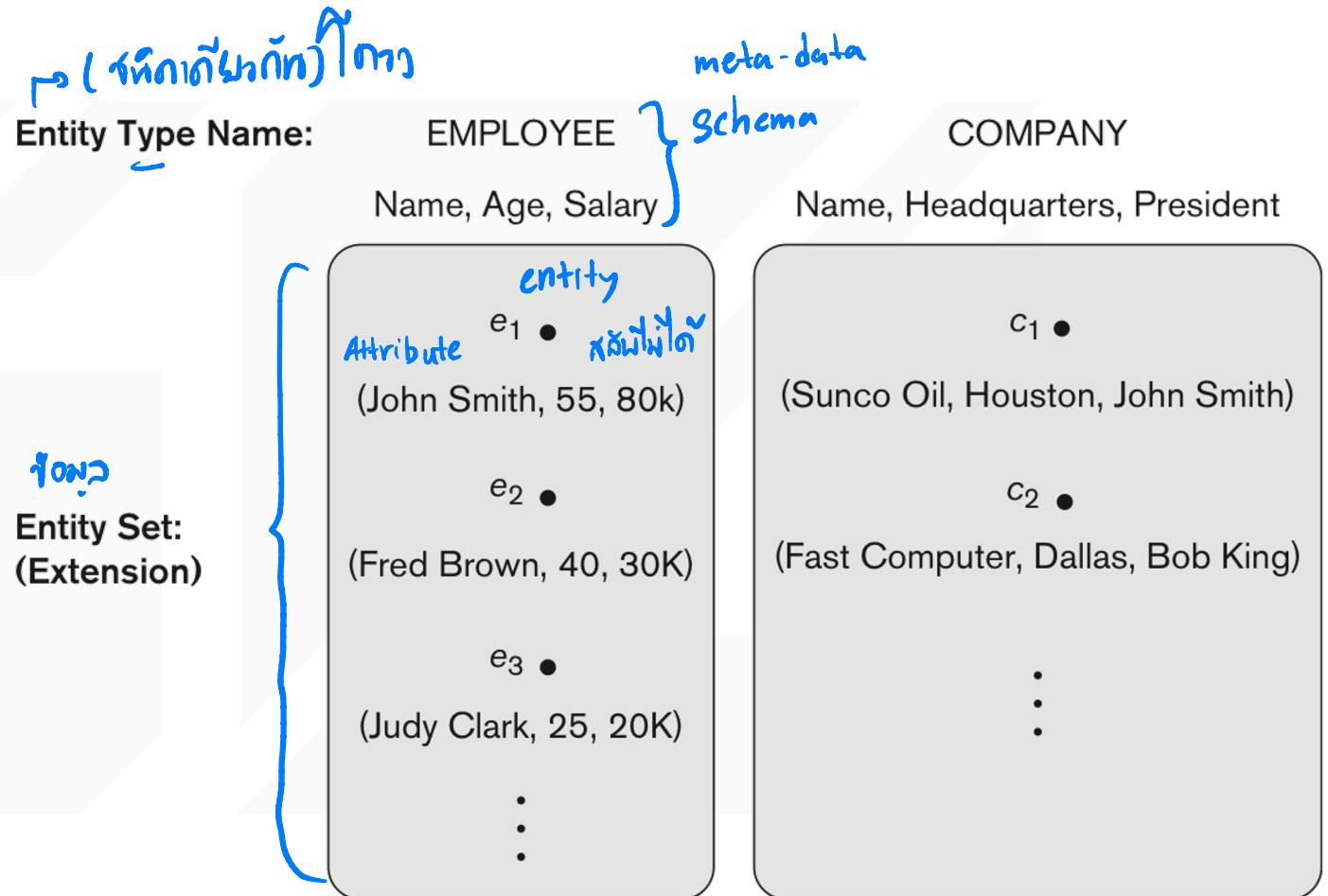
## • NULL value

- Not applicable មិនអាចប្រើបាន
- Unknown មិនដឹង

  - The attribute value is missing or
  - Not known

# Entity Types

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



# Entity Sets

- Each entity type will have a collection of entities stored in the database
  - Called the **entity set** or sometimes **entity collection**
  - Same name used to refer to both the **entity type** and the **entity set**
  - However, entity type and entity set **may be given different names**
  - Entity set is **the current state of the entities of that type** that are stored in the database

(Attribute များ Null ထု / Nullမှုနဲ့)

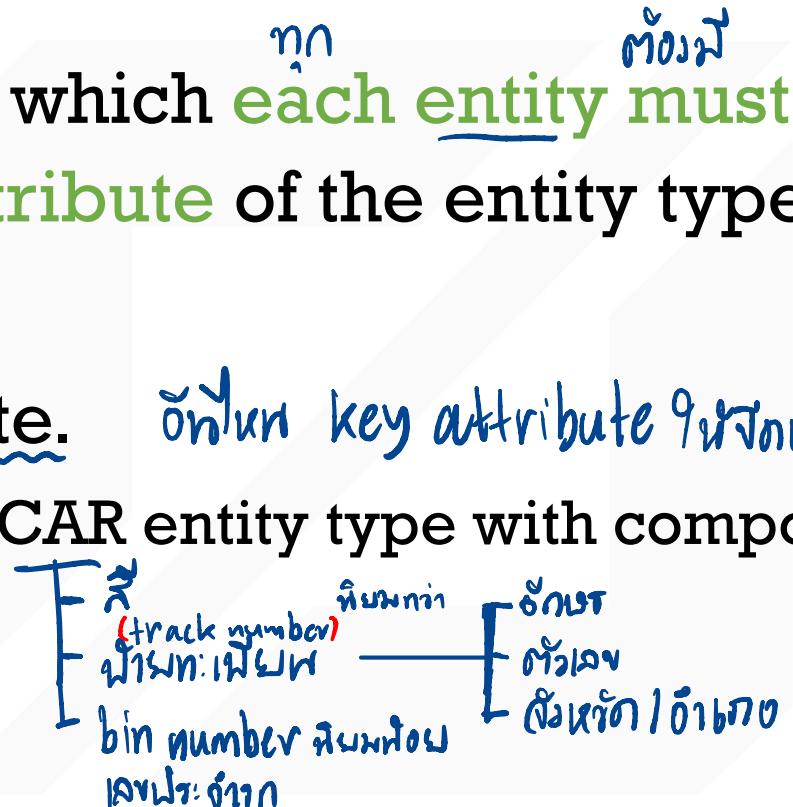
# Key Attribute

→ ទីនុការណ៍វិត្ត ក្នុងការព័ត៌មាន "រាយ Entity វិត្ត" | ចាប់បើ

↳ **Attributes** | key Attribute

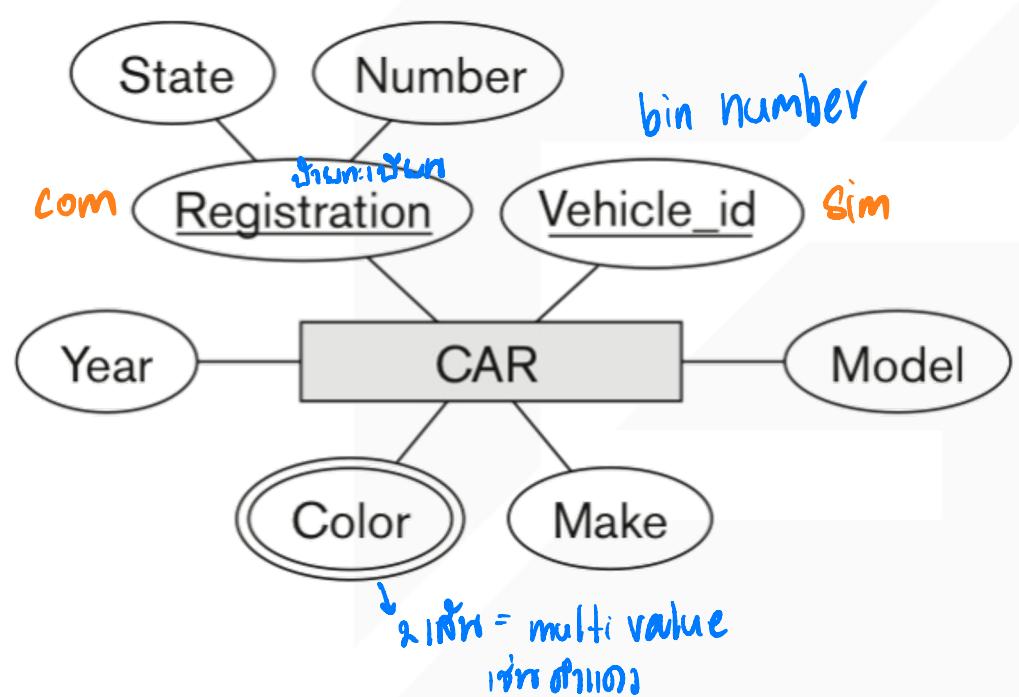
- **Entity** මාන  
- **entity** value

- An attribute of an entity type for which each entity must have unique value is called a **key attribute** of the entity type.
    - For example, SSN of EMPLOYEE.
  - A **key attribute** may be composite.
    - VehicleTagNumber is a key of the CAR entity type with components (Number, State).

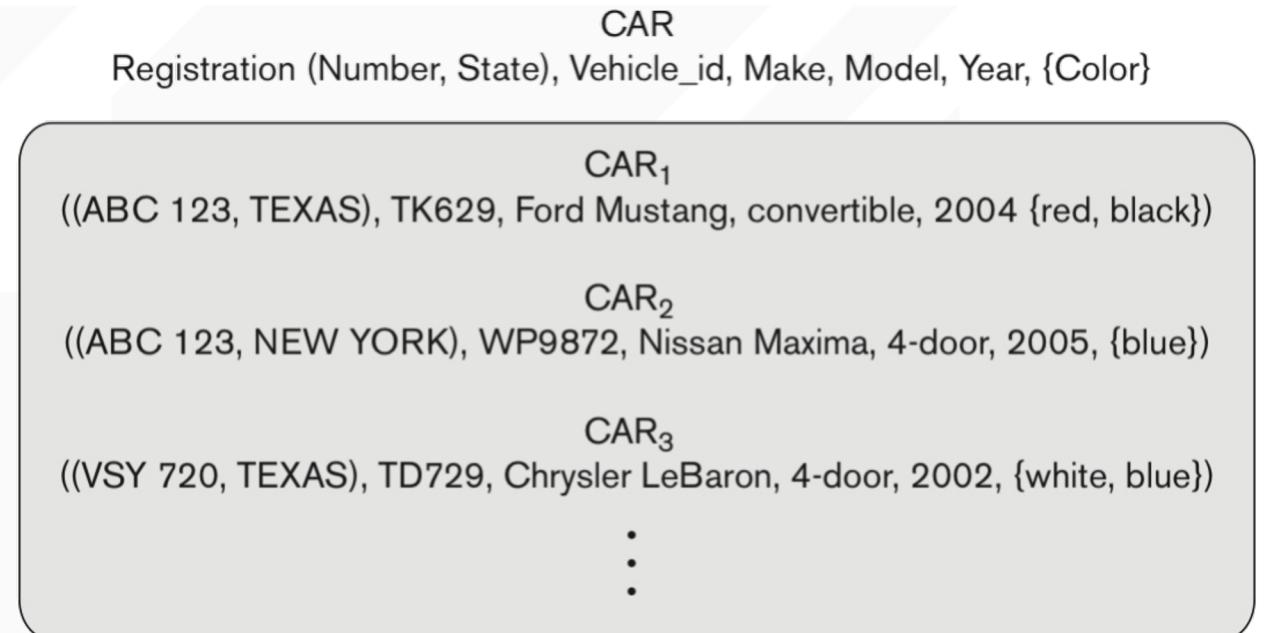


- An entity type may have more than one key.
  - The CAR entity type may have two keys:
    - VehicleIdentificationNumber (popularly called VIN)
    - VehicleTagNumber (Number, State), aka license plate number
- Each key is underlined ອັນກົມ key attribute ຖະຈາວທີ່ກຳນົດ
- Note: this is different from the relational schema where only one “primary key” is underlined.

ຫຼັງຈາກ key ສິນ Er ຈຳເປັນຕົວ ດີ ກຳນົດ (ຈິຕູງຕົວ)



The CAR entity type with two key attributes; Registration and Vehicle\_id.



Entity set with three entities

# Displaying an Entity type

- In ER diagrams, an **entity type** is displayed in a **rectangular box** 
- **Attributes** are displayed in **ovals** 
  - Each attribute is connected to its entity type
  - Components of a composite attribute are connected to the oval representing the composite attribute
  - Each key attribute is **underlined**
  - Multivalued attributes displayed in **double ovals**

**Figure 3.14**  
Summary of the notation for ER diagrams.

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship <i>Week</i>
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute <i>คำนวณได้</i>
	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 $R$

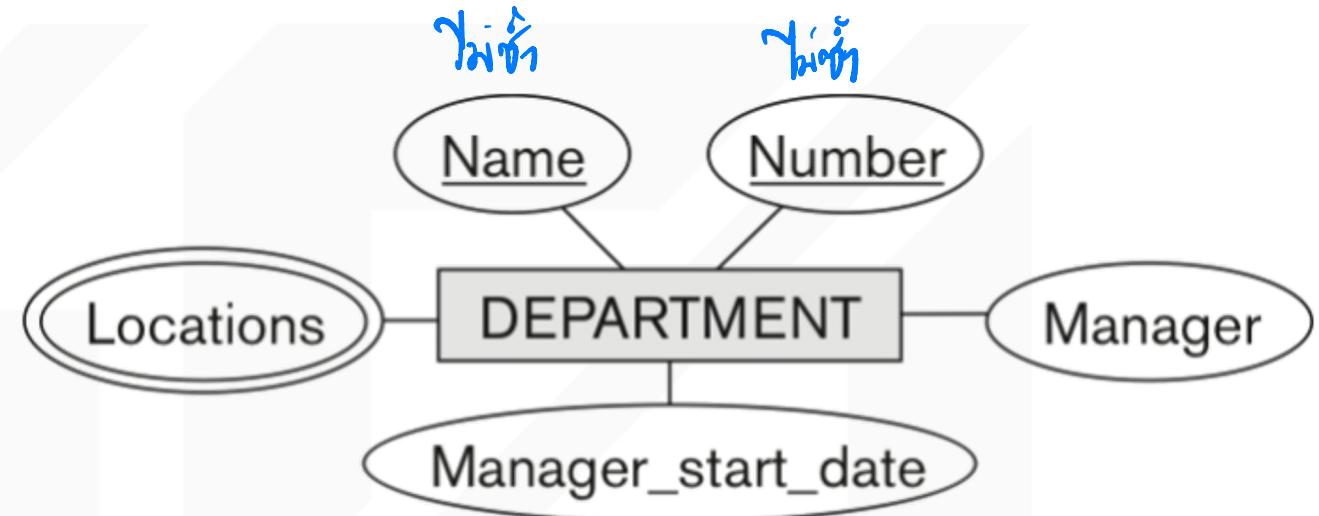
# Initial Conceptual Design

- The company is organized into **DEPARTMENTS**.

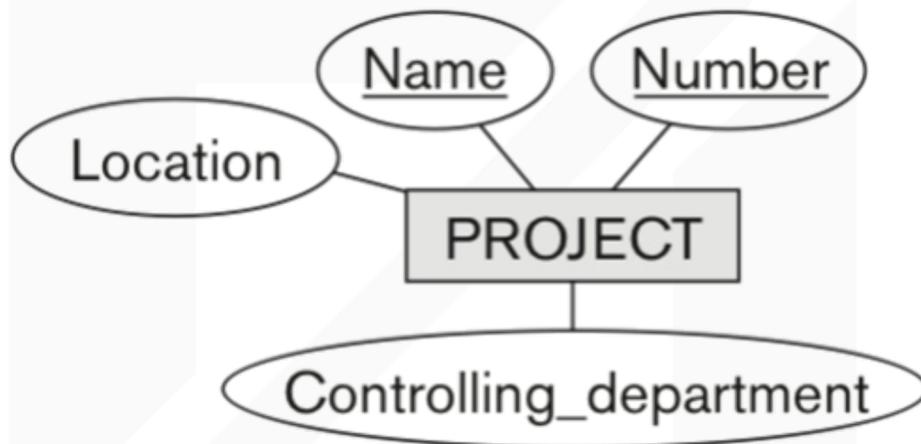
Each department has a **name**, **number** and an **employee** who **manages** the department.

We keep track of the **start date** of the **department manager**.

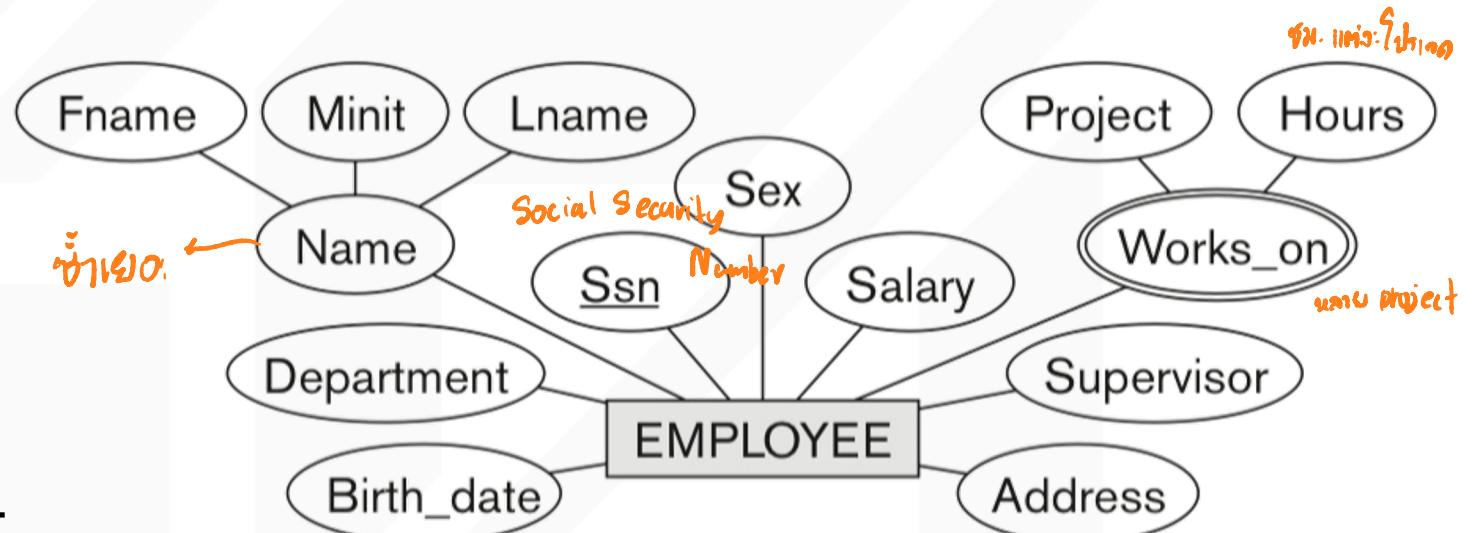
A department may have **several locations**. *multi value*



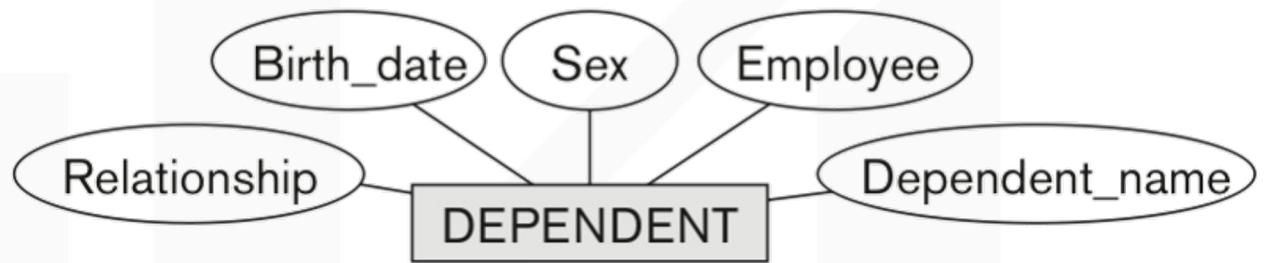
- Each **department** controls a number of **PROJECTs**
- Each project has a **unique name**, **unique number** and is located at a single location.



- The database will store each EMPLOYEE's social security number, address, salary, sex, and birthdate.
- Each employee works for one department but may work on several projects.
- The DB will keep track of the number of hours per week that an employee currently works on each project.
- It is required to keep track of the direct supervisor of each employee



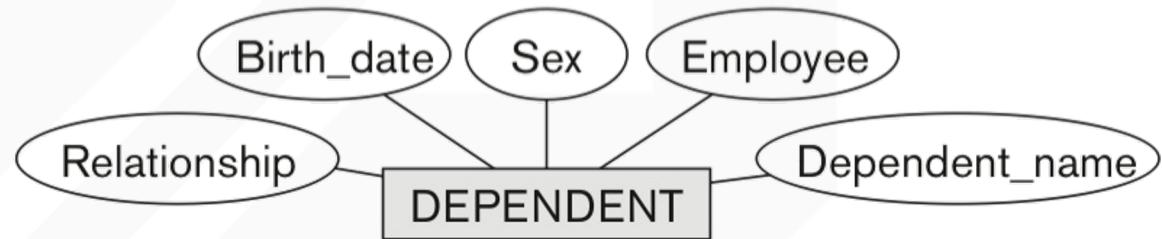
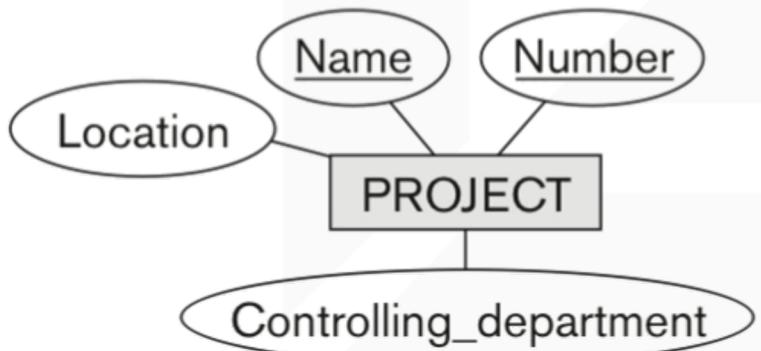
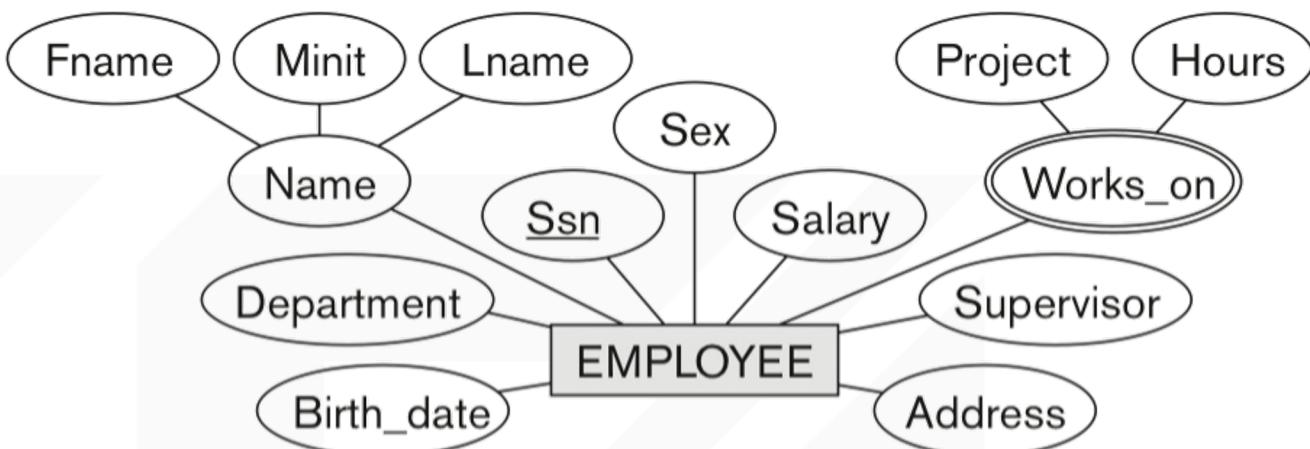
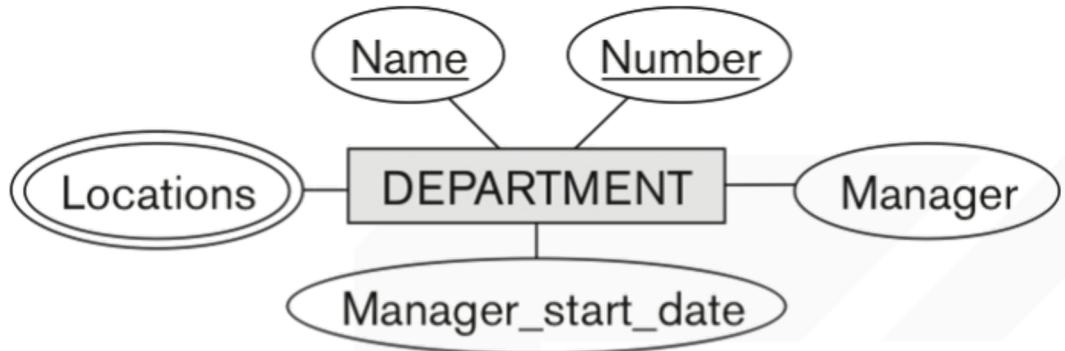
- Each employee may have a number of **DEPENDENTS**.
- For each dependent, the DB keeps a record of **name**, **sex**, **birthdate**, and **relationship** to the **employee**



# សម្រាប់បង្កើតការណ៍ព័ត៌មាន នៃក្រុមហ៊្រកិន "អ៊ីនី" ដែលមានពាណិជ្ជកម្ម

1816 63010179

ឈ្មោះ: លោកស្រី នាមី កែវិសាល ឯទាឡ 63050777 ឯទាឡ



- **ER model has three main concepts:**

- Entities (and their entity types and entity sets) *object ទីរាង*
- Attributes (simple, composite, multivalued) *ព័ត៌មានជាមុនភាព* *\*Attribute ព័ត៌មាន*
- Relationships (and their relationship types and relationship sets) *សមារៈព័ត៌មាន* *Entity*

# Relationships and Relationship Types

១. សំណើន៍ ២. ពិភ័យ

- 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**.

Attributed

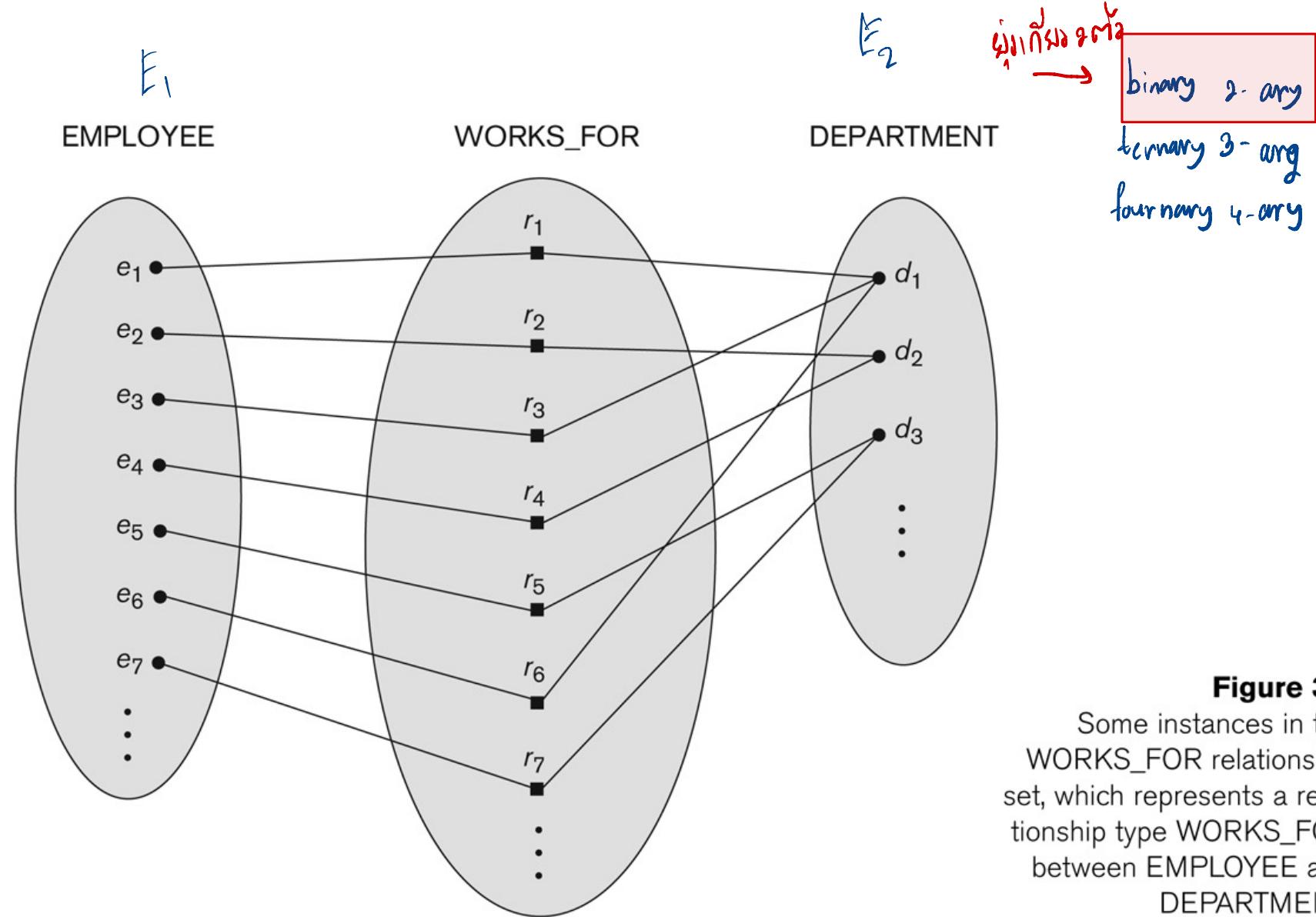
Attributed

២. សំណើន៍រាយ Employee នូវ project និង works on  
និង Attributable

- 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.

ការពារ Entity ទីមួយកែលចាក្រា

- The **degree** of a relationship type is the number of participating entity types.
  - Both MANAGES and WORKS\_ON are **binary** relationships



**Figure 3.9**  
Some instances in the WORKS\_FOR relationship set, which represents a relationship type WORKS\_FOR between EMPLOYEE and DEPARTMENT.

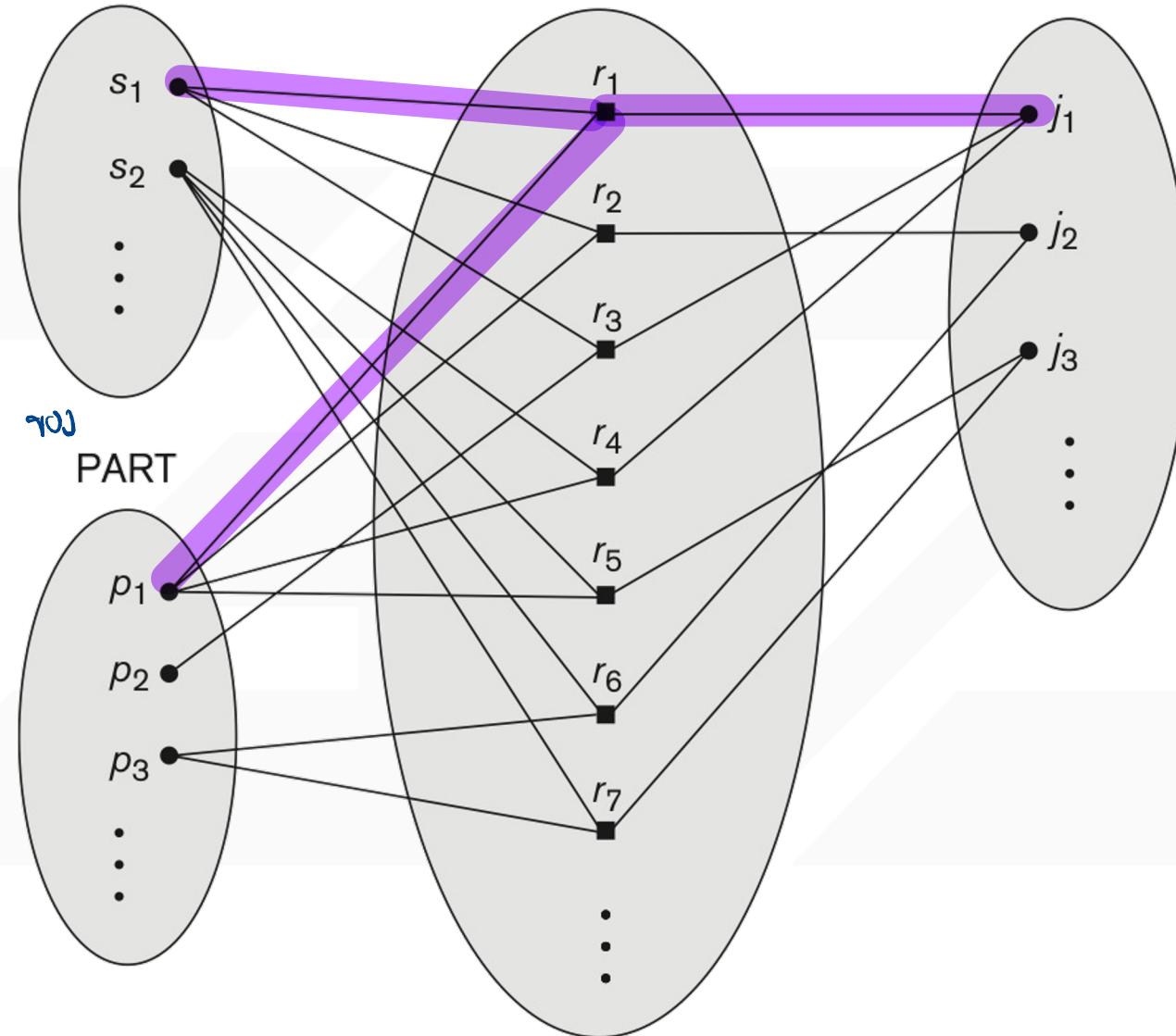
## • Degree of a Relationship Type

- The number of participating entity type.
  - E.g., the WORKS\_FOR relationship is of degree two
- Degree of two is called binary
- Degree of three is called ternary.

ການສ່ວນ  
SUPPLIER      SUPPLY      PROJECT

Entity

ໃຫຍ່ Ternary  
3-ary



## • **Recursive Relationship Type**

- A relationship type between the same participating entity type in **distinct roles** ร่องรอย / รูปแบบ
- Also called a **self-referencing** relationship type.
- Example: the SUPERVISION relationship

- 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 ເຕັກຍ

- One employee in *supervisee* role ລູກໍຈາ (ຜູ້ນິ້ນຕົກກາ) / ຜູ້ດູກ

ມັນຍາ

ការអនុវត្ត ឱ្យរាយ ការអនុវត្ត

កំណត់ការ → នគរបាលការកំណត់

- In a **recursive relationship** type.

- Both participations are same entity type in different roles.

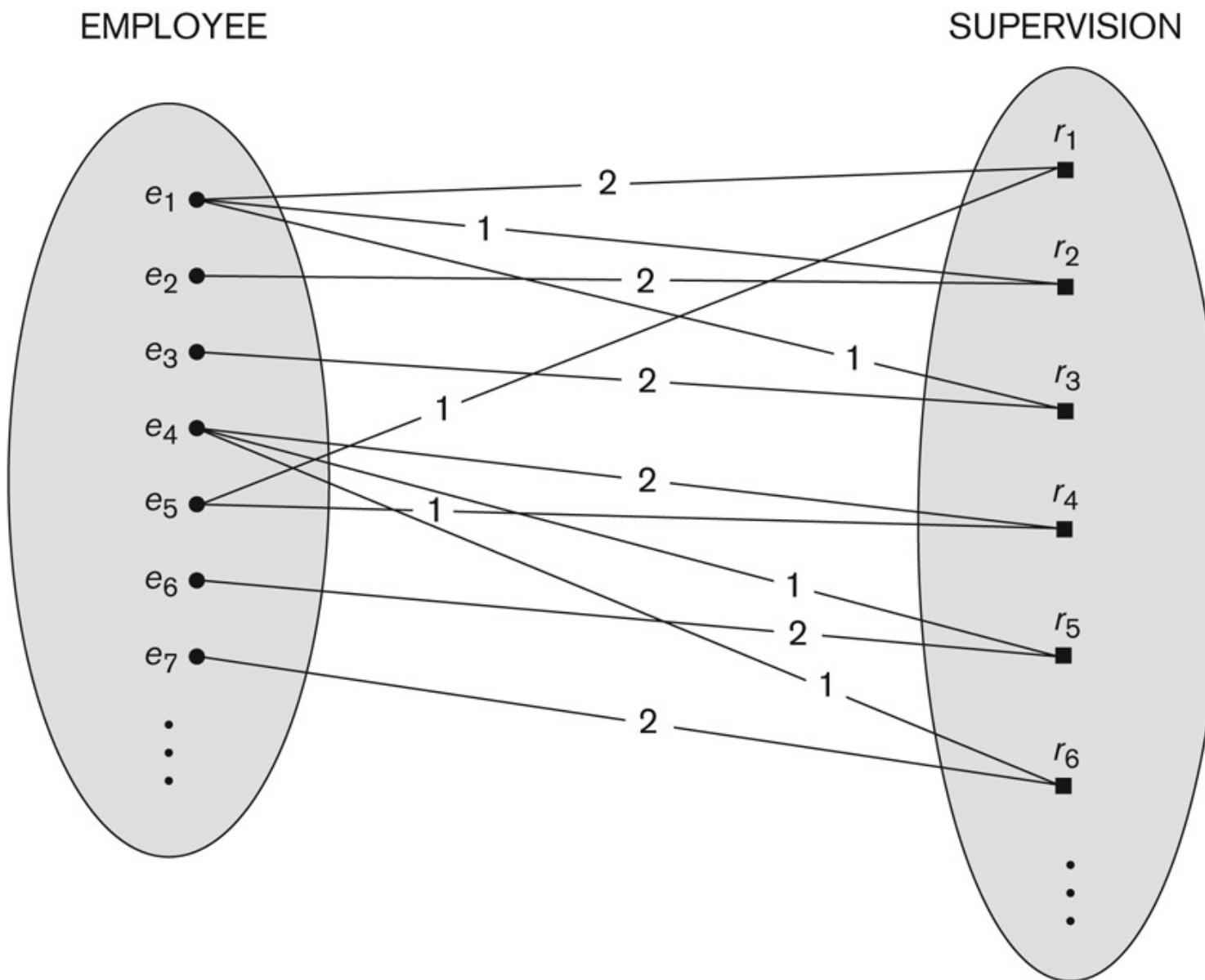
- For example,

SUPERVISION relationships between EMPLOYEE  
(in role of supervisor or boss)

and

(another) EMPLOYEE  
(in role of subordinate or worker).

2 sub  
1 supervisor



**Figure 3.11**

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

# 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

} relationship mapping

- 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** →  $\rightarrow_E$
  - Different meanings and different relationship instances

→  $E$

2 E อยู่ 2 Relate

ឧទ្ទិន

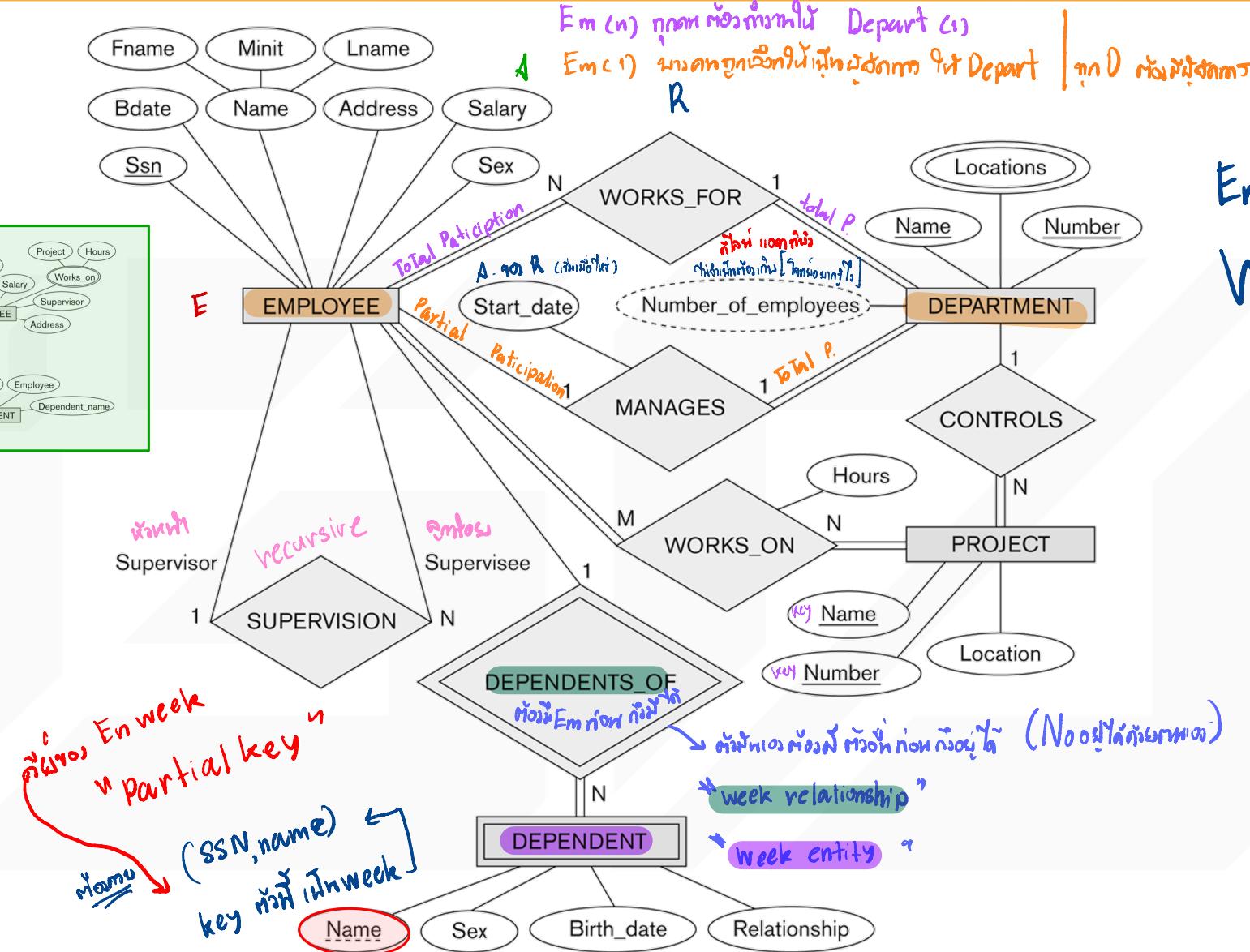
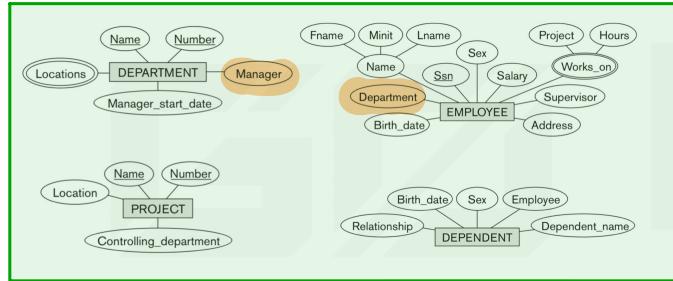


Figure 3.2

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

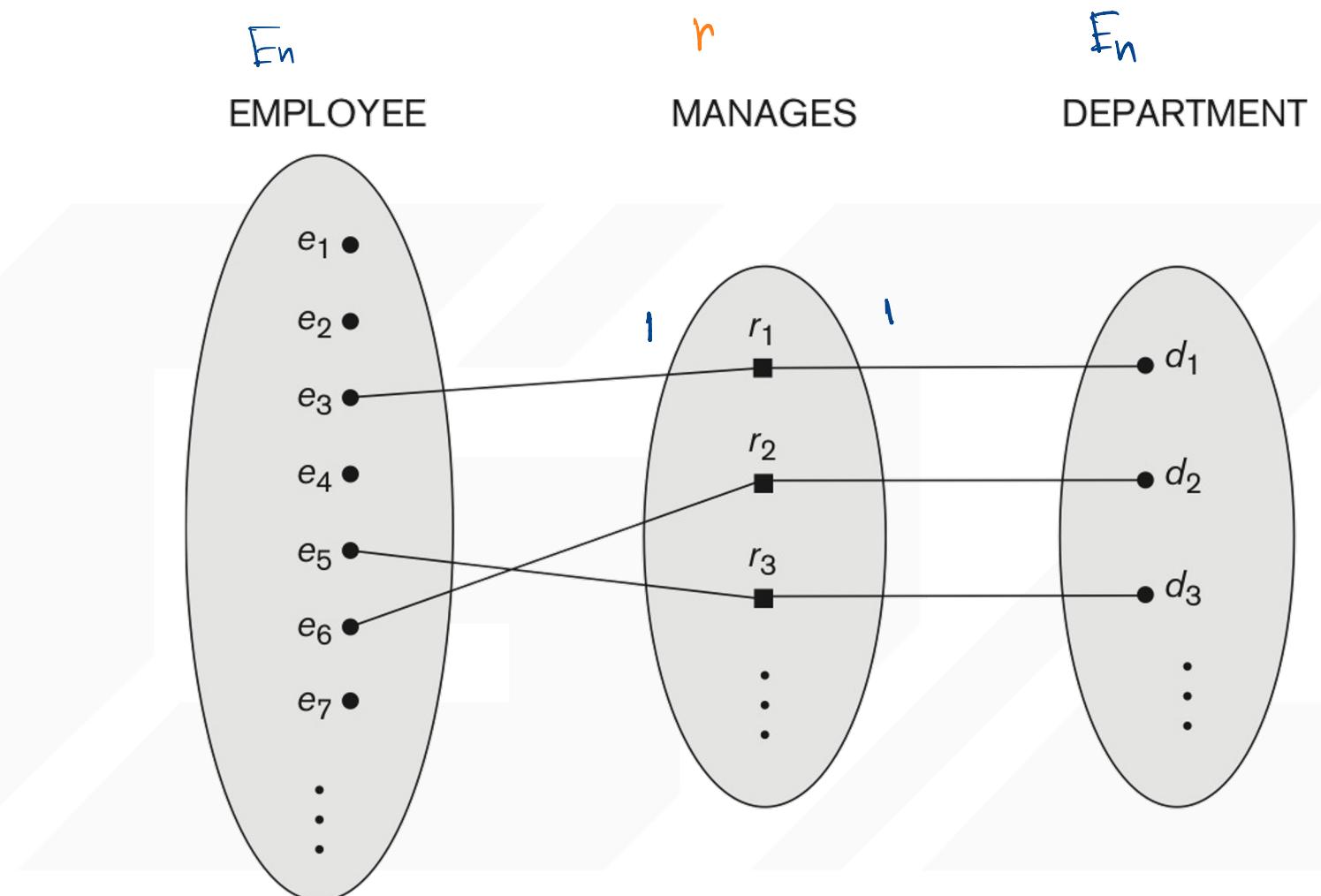
En/Re

En/Re  
Week — strong

# Constraints on Relationships

- **Constraints on Relationship Types**

- Also known as ratio constraints **ตัวกำหนด Max** “**ค่าความมากที่สุด**”
- **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)



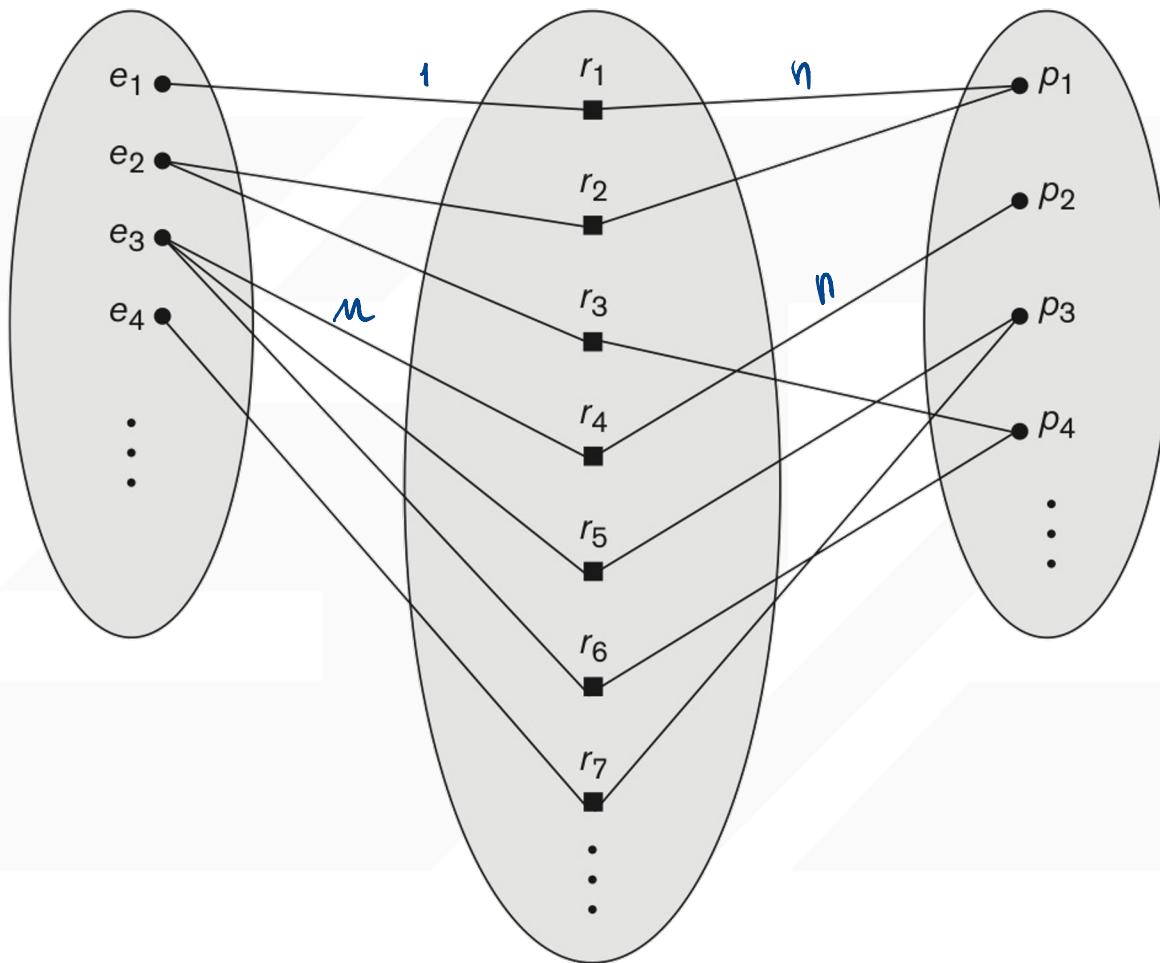
A 1:1 relationship, MANAGES

## ពិភាក្សា ការងារក្នុង

EMPLOYEE

WORKS\_ON

PROJECT



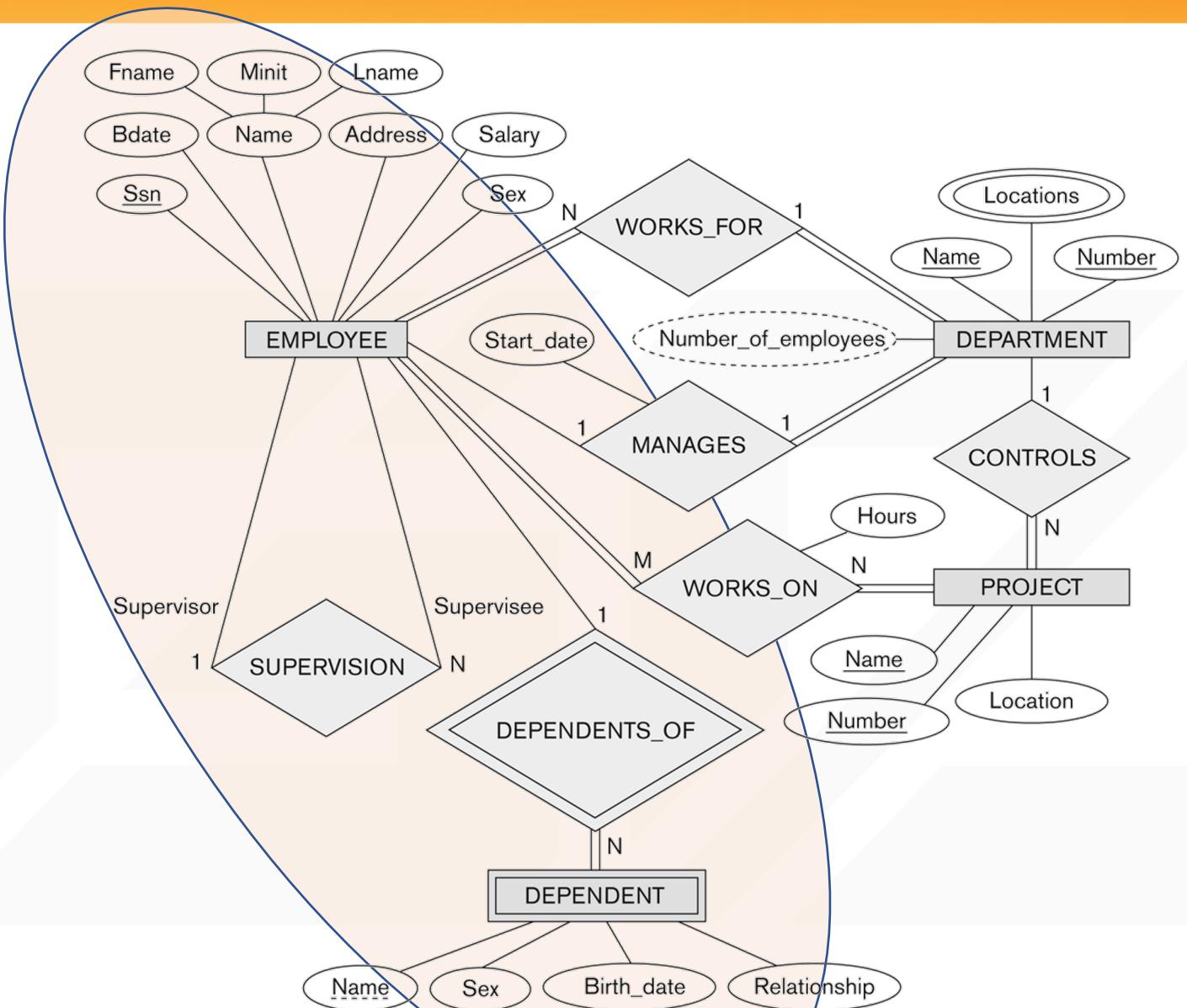
An M:N relationship, WORKS\_ON

# Weak Entity Types ព័ត៌មានអង្គភាព

- An entity that **does not have a key attribute** and that **is identification-dependent on another entity type**.
- 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 relationship 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**

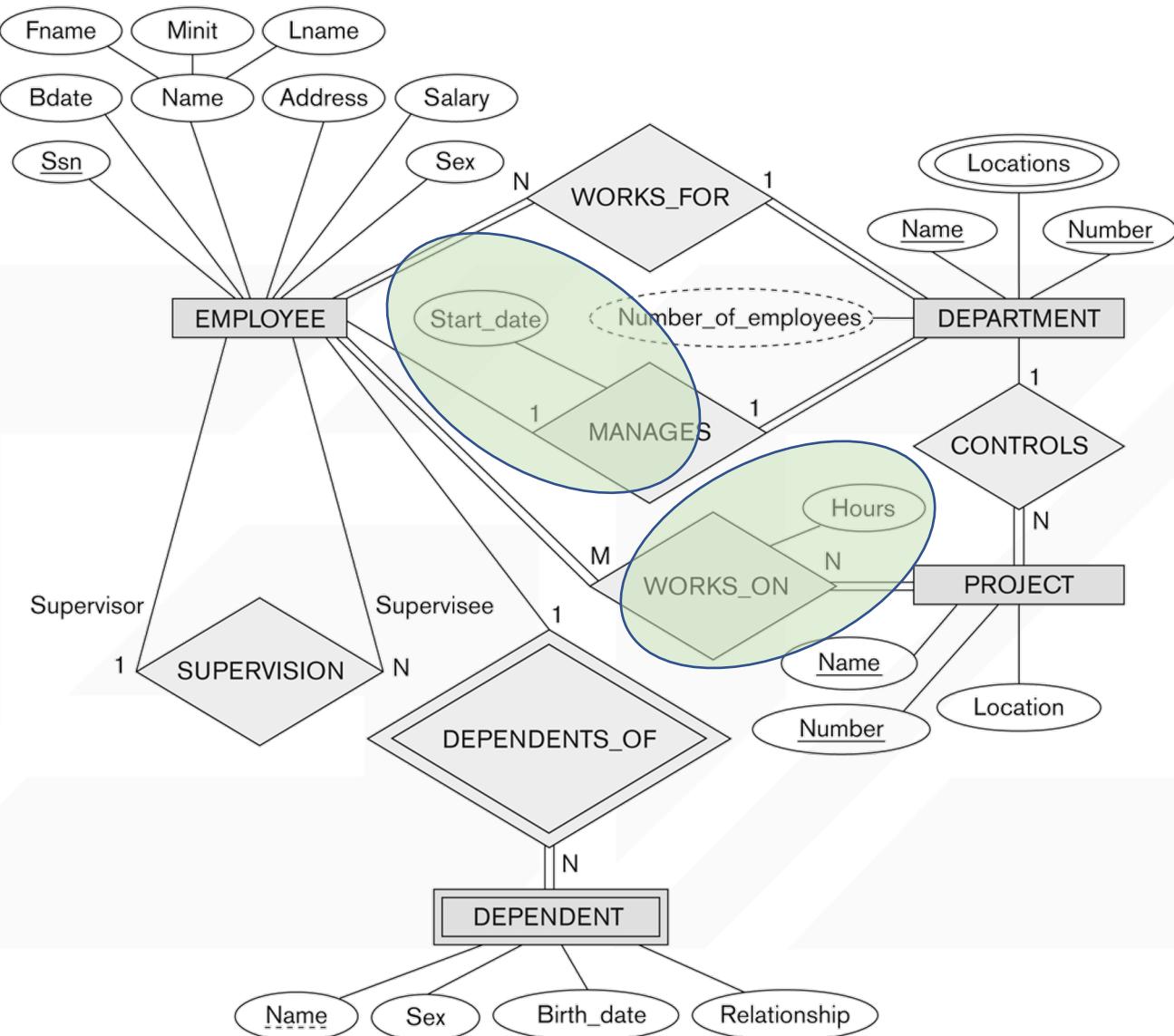


**Figure 3.2**

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

# Attributes of Relationship Types

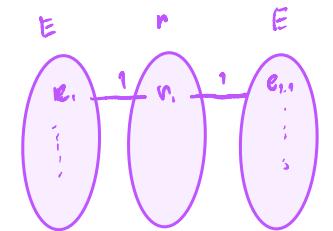
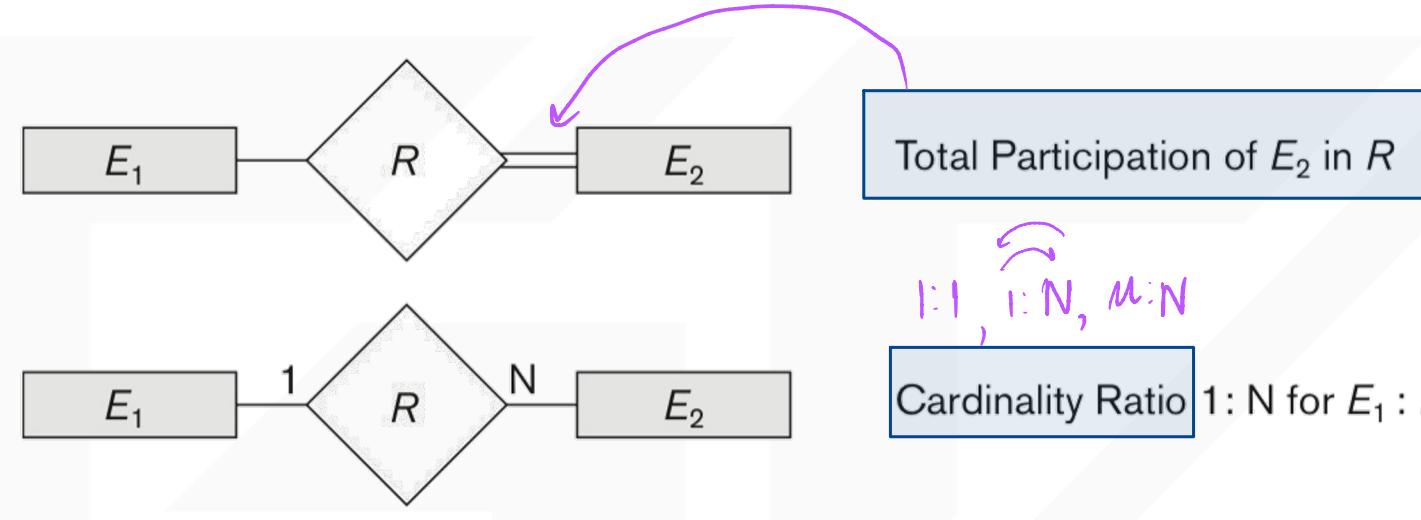
- A relationship type can have attributes:
  - For example, **HoursPerWeek** of **WORKS\_ON**
  - Its value for each relationship instance describes the number of hours per week that an **EMPLOYEE** works on a **PROJECT**.
    - A value of **HoursPerWeek** depends on a particular (employee, project) 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



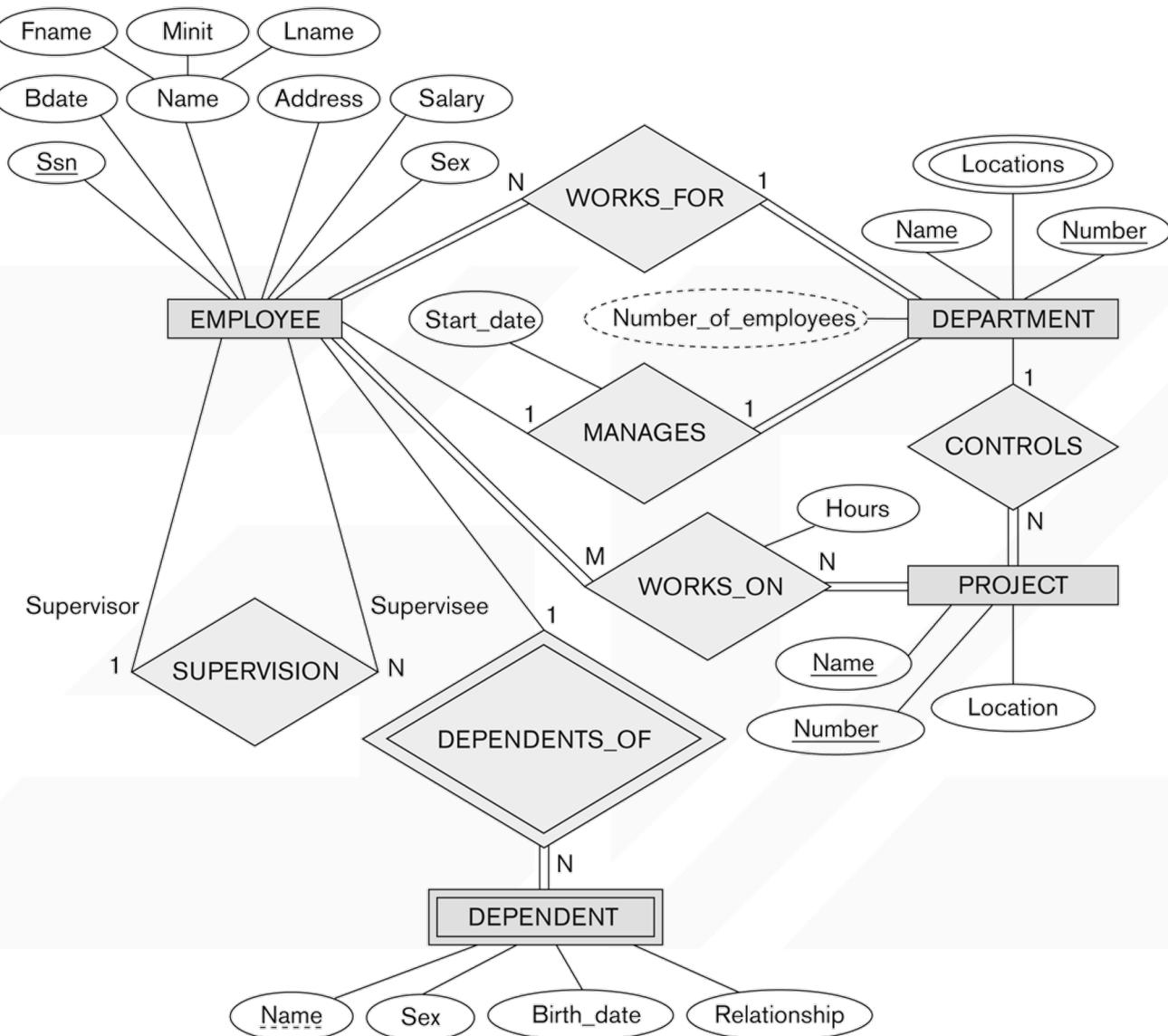
**Figure 3.2**

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.



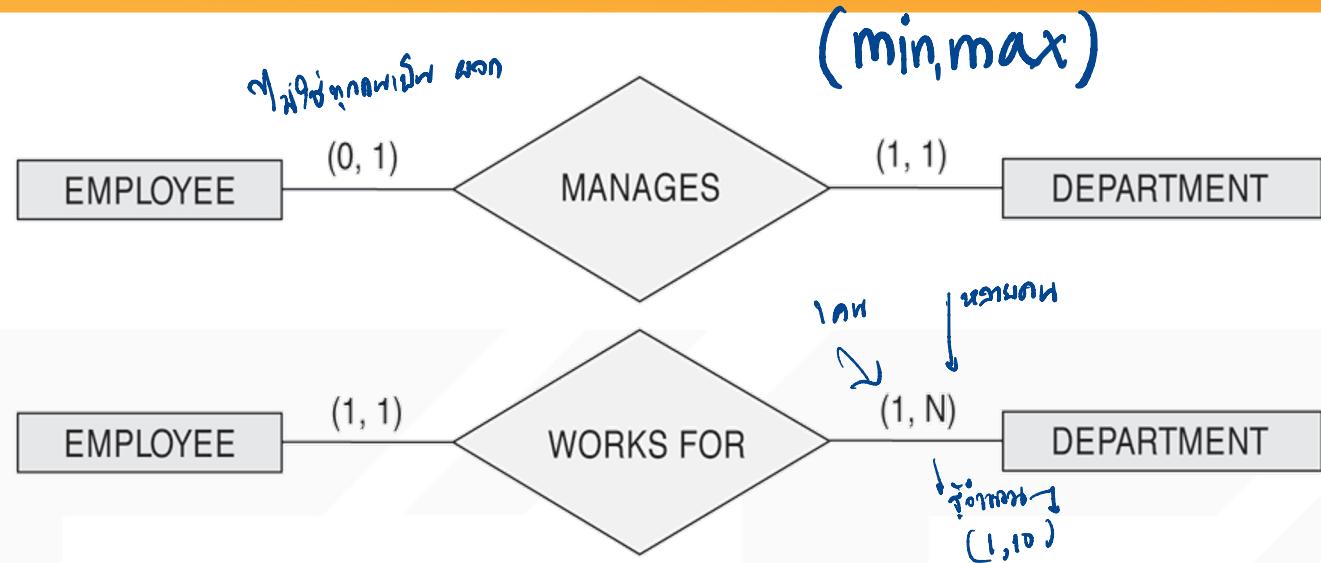
**Figure 3.2**

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

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

กำหนด min, max สำหรับ

- 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):  $\text{min}=0, \text{max}=n$  (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

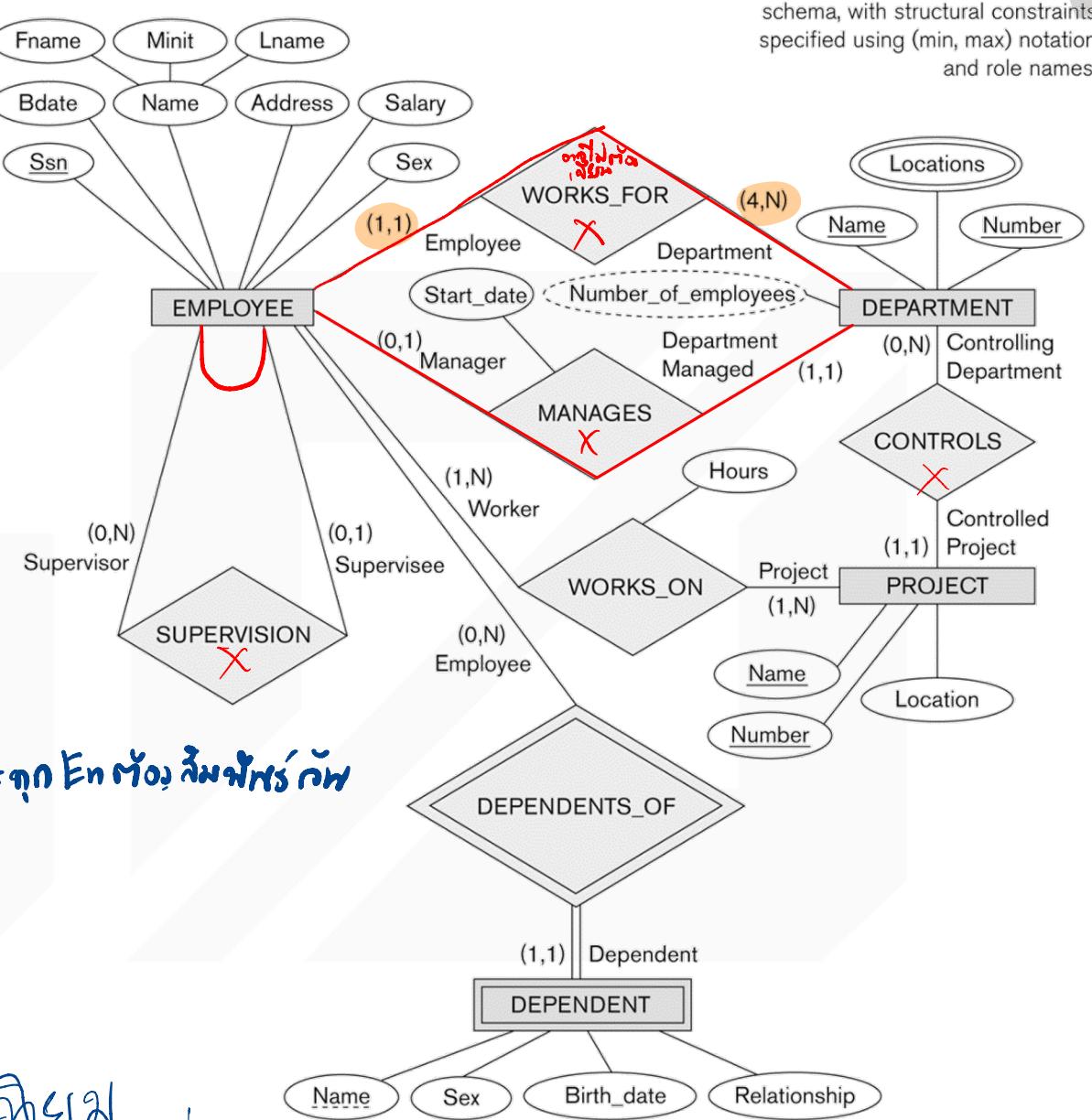
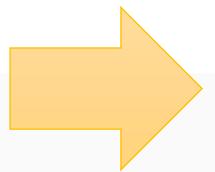
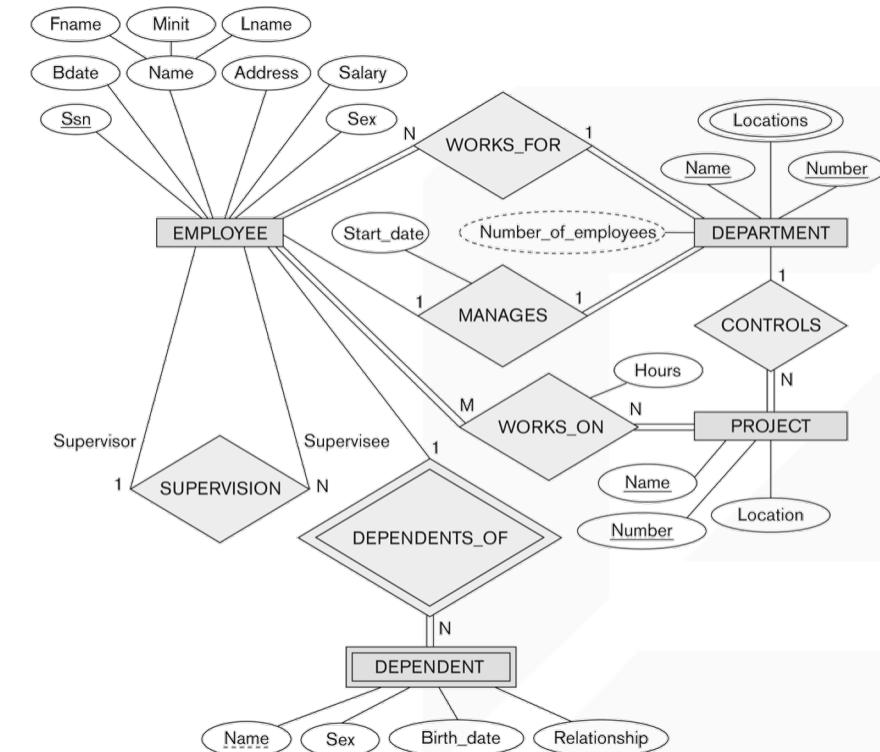


## • Examples:

- 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

Figure 3.15

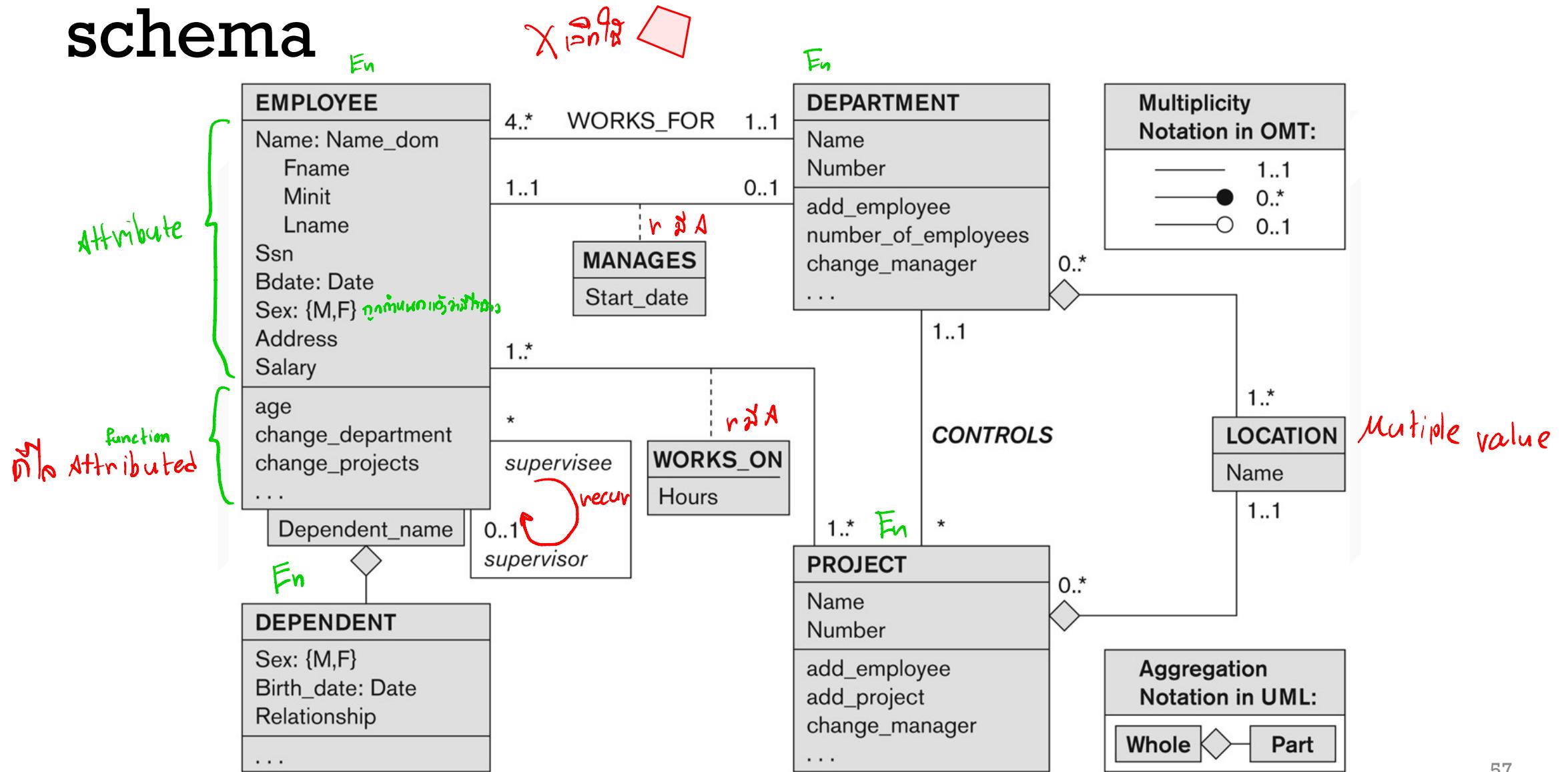
ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.



# 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

# UML class diagram for COMPANY database schema



# ER model

## entities

ទម្រង់ (ក្នុង: en)

- En Type -  
Attri និរនោតនាពី
- "few entities គឺជាដំឡើង" Entity set
  - Entity Collection
  - or formular
- week En នឹង participate / Identifying en
  - ↳ partial key = នូវយ៉ាងដែលសម្រាប់

## Attribute

### simple

### composite (..)

### multi value {...}

### store vs derived

### null value null

### key Attribute

(និរនោត)  
A ឬ A

(និរនោត)  
A ឬ En

value ក្នុងអាជីវការ  
ឬក្នុង

ឬក្នុង  
ទិន្នន័យ  
ក្នុង Entities នៃខ្លួន

## relationship

### attribute នៃ

និរនោត និងការពារ 2 Entity

re... type → same type Ex. En ពេលកំណត់

degree n. En នឹង re នូវ en. —————— en<sub>1</sub> ឬ en<sub>2</sub> ឬ en<sub>3</sub> 3-ary

binary 2-ary

ternary 3-ary

recursive ត្រូវការកំណត់ En ពេលកំណត់ (En employee)

En ដែលត្រូវការកំណត់ និងត្រូវការកំណត់

partial participation [total participation]  
partial participation /

Constraints នៃការកំណត់ n. និងការកំណត់ Cardi max 1:N, M:N, 1:1

option  
non option (min, max)

Existence min

→ ក្នុងការកំណត់ការកំណត់ UML

UML class diagram for COMPANY database schema

