

Database Systems

Program in Computer Engineering
School of Engineering

King Mongkut's Institute of Technology Ladkrabang

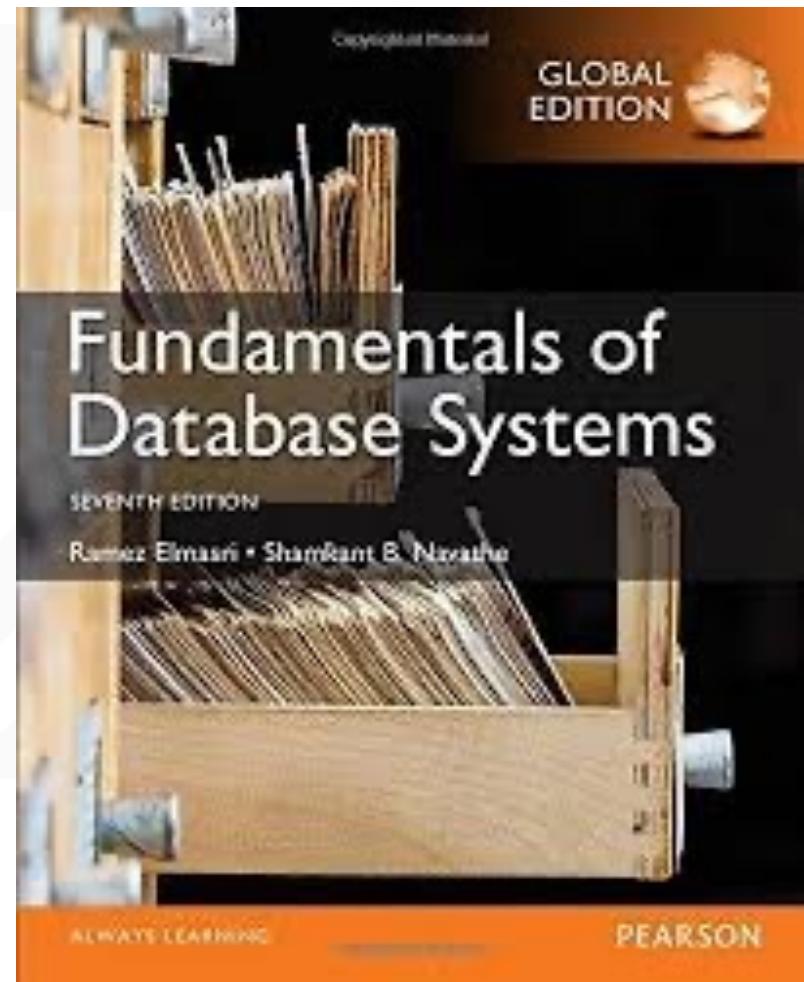
Database Systems

Program in Computer Engineering
School of Engineering

King Mongkut's Institute of Technology Ladkrabang

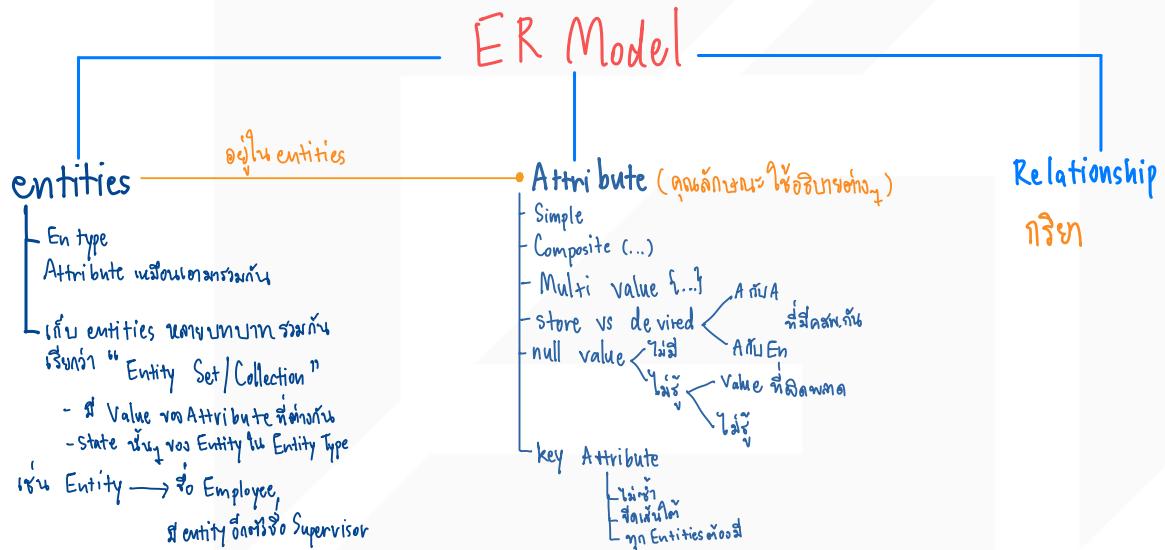
Text

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



Chapter 3

Data Modeling Using the Entity-Relationship (ER) Model



เอกสารนี้มีจุดเด่นดังนี้

- การอธิบาย ER Model อย่างง่ายๆ
- ความสัมภาระของ Entity, Attribute และ Relationship
- การอธิบายประเภทของ Attribute (Simple, Composite, Multi-value, etc.)
- การอธิบาย Cardinality และ Multiplicity ใน Relationship

เอกสารนี้ช่วยให้เข้าใจโครงสร้างข้อมูลใน ER Model ได้ดีขึ้น

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

Programme

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

Diagram
ដំឡើងនៃការរចនាខ្សែសម្រាប់បណ្តុះបណ្តាល

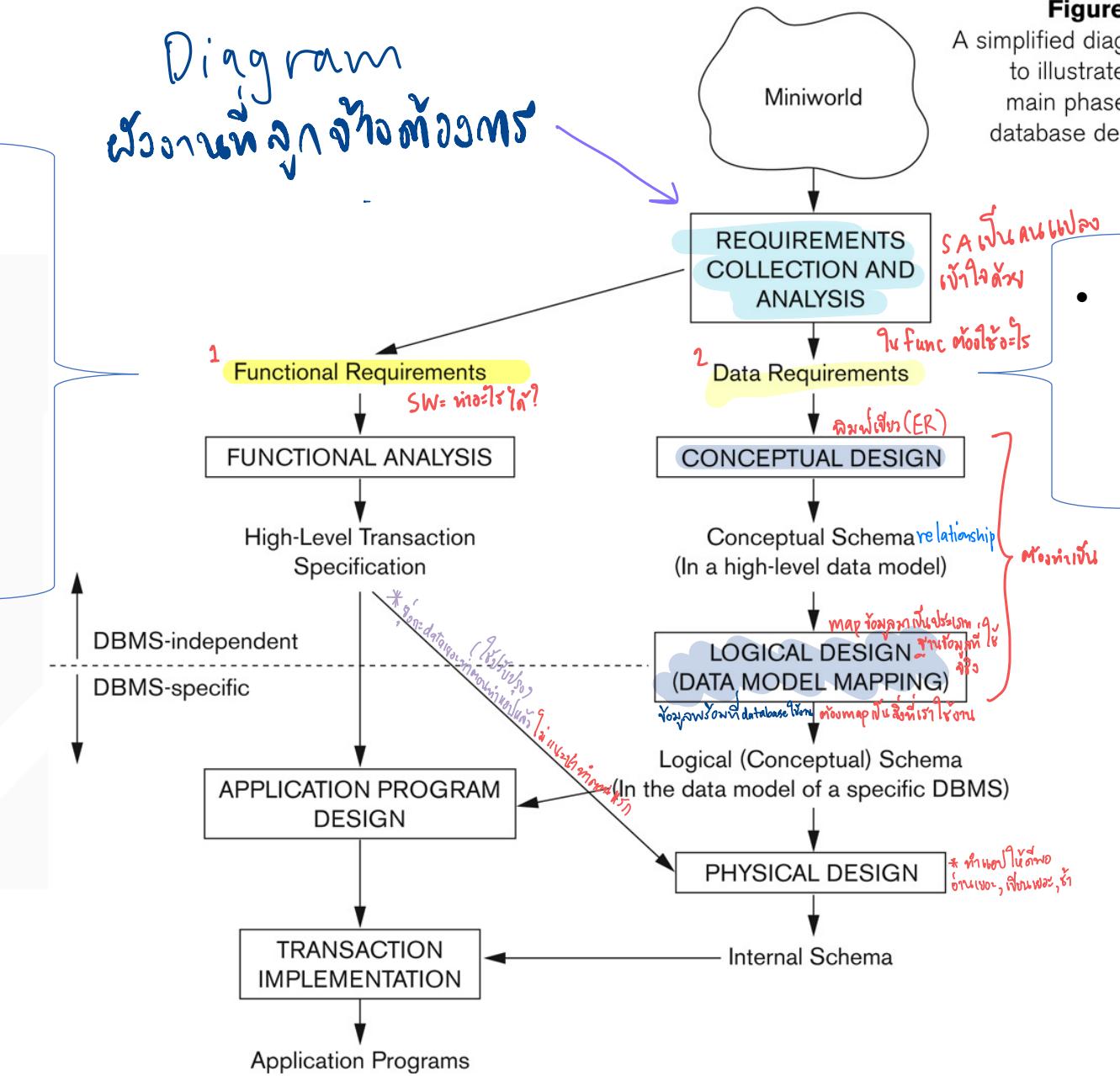


Figure 3.1

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

សាកលវិទ្យា
គ្រប់គ្រង

- Detailed description
 - Entity types
 - Relationships
 - Constraints



ឧប្បរិយា

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 ¹a name, ²number and an ³employee who manages the department.

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

A department may have several locations.

- Each department controls a number of PROJECTS.
Each project has ¹a unique name, ²unique number and is located at a single location.

ឈ្មោះ Project ឈ្មោះ លេខកូដ នាមឈ្មោះ = ចំណាំលក្ខណៈ

- 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

SSN

ประวัติส่วนบุคคล
ที่อยู่บ้าน

(หมายความว่า งานที่ทำ)

เงินเดือน/เดือน
รายเดือน

1 เดือน 2 เดือน

โครงการ
Project

ชั่วโมงทำงานต่อสัปดาห์

ผู้ดูแลพนักงานเด็ก

ผู้ดูแลคนเด็ก

ជីវិស័យសាធារណរដ្ឋមន្ត្រី ក្រសួងការងារ

ស្នូល | ក្រសួង | ឯកសារ | អគ្គនាយក

- 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

Entities and Attributes

(n.)

/entity
entity

คุณลักษณะ

- 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

- Attributes are properties used to describe an entity.

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

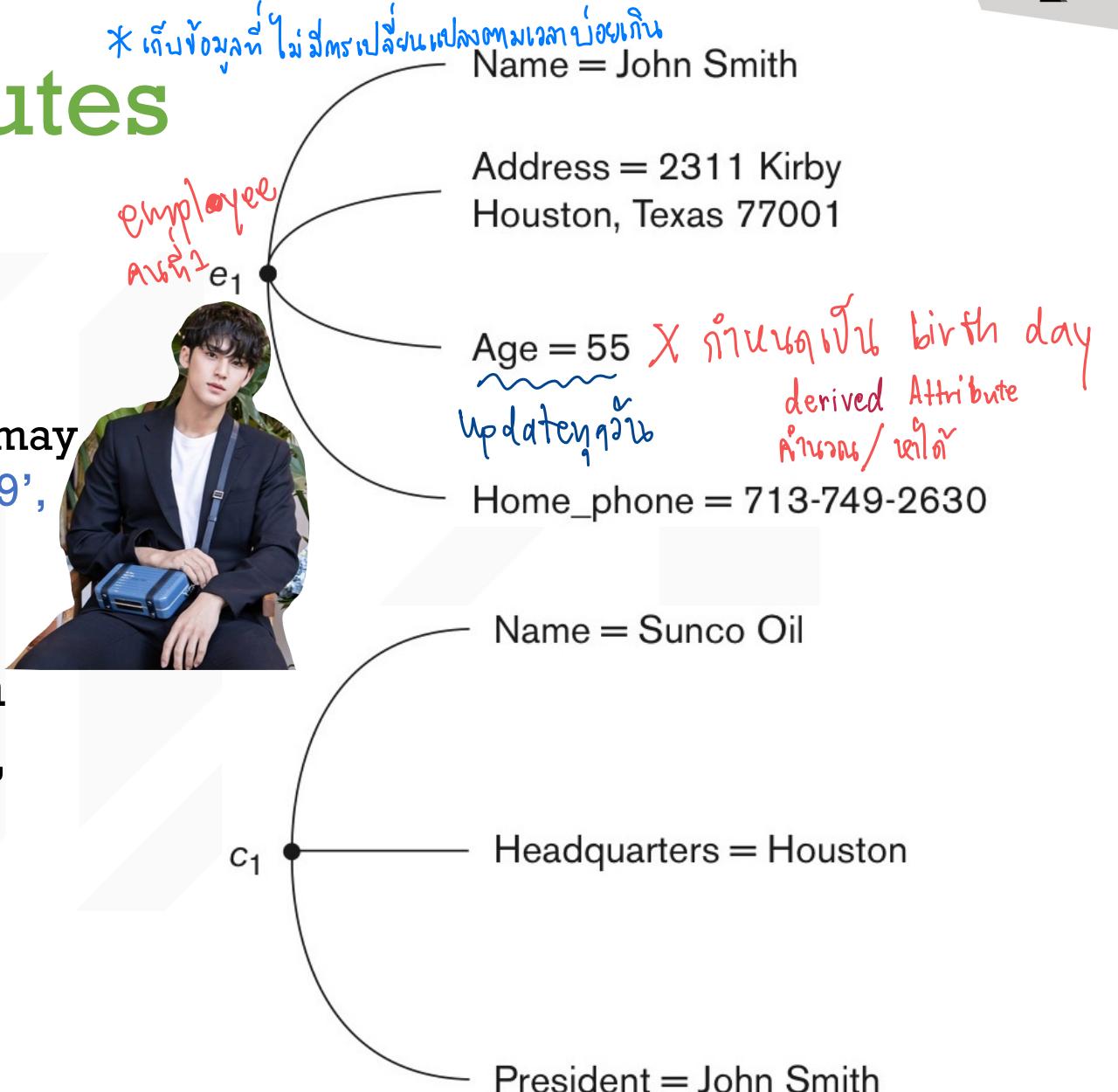
Entities and Attributes

- 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, ...

ក្នុងចំណាំមានតម្លៃខ្លួន



Types of Attributes

- **1 Simple**

- Each entity has a single atomic value for the attribute.
- For example, SSN or Sex. *ເລີຍບົນຫຼວມຕະໂລກ*

- **2 Composite** *ເອກຕາມ, ປະເຈດ ຫຼື ຊິ້ນຈຳນວດກັນ*

- The attribute may be composed of several components.
- For example: *ສິ່ງເລັບ(..)*
 - Address(Apt#, House#, Street, City, State, ZipCode, Country), or
 - Name(FirstName, MiddleName, LastName).
 - Composition may form a hierarchy where some components are themselves composite.

- **3 Multi-valued**

Attribute ສືບຕົກນທຸກຄູ່

- An entity may have multiple values for that attribute.
- For example, *ສິ່ງໃໝ່{..}*
 - Color of a CAR or *ຮອນພົນ*
 - PreviousDegrees of a STUDENT. *ມູນລົງລາຍແບບ*
 - Denoted as {Color} or {PreviousDegrees}.

ចំណាំការពារ

លោកស្រីជាមួយនឹងវគ្គលេខា
កំ

- 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

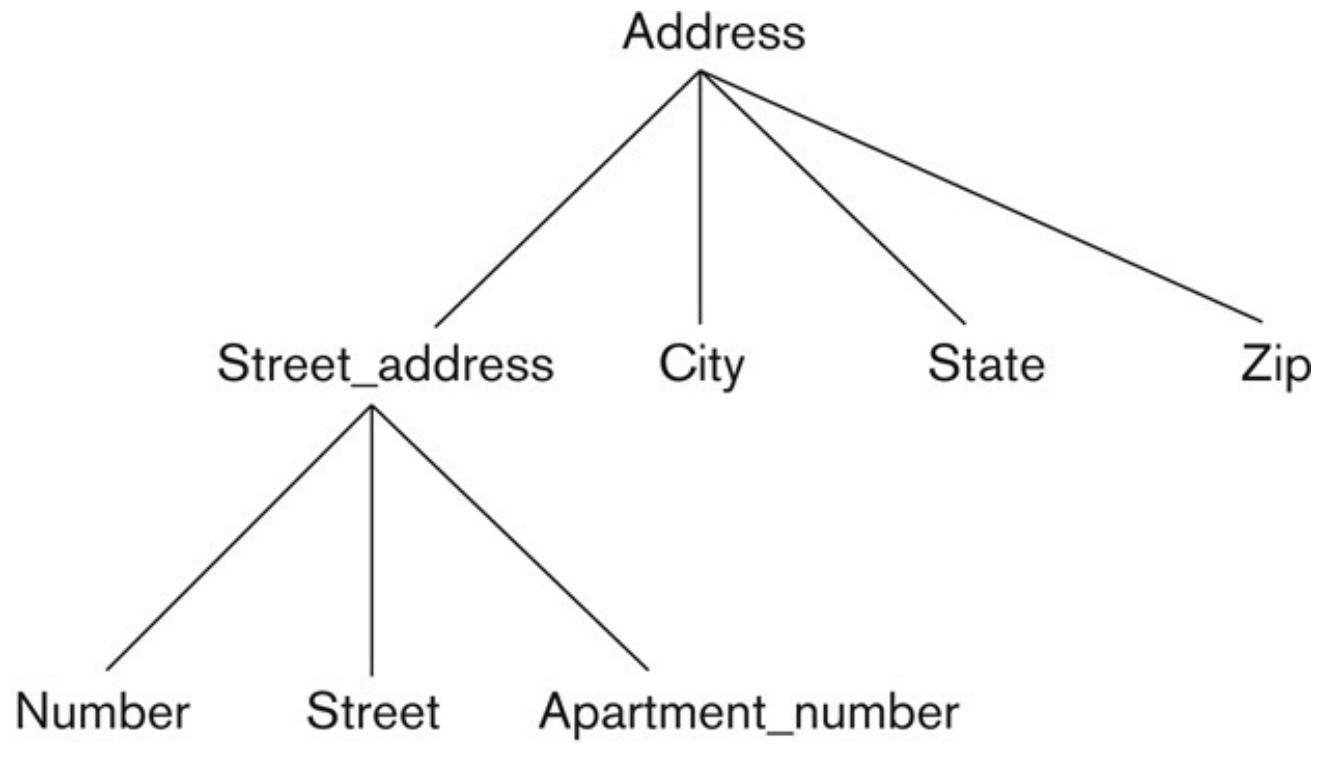


Figure 3.4
A hierarchy of composite attributes.

- Attribute ជំនួយក្នុងបច្ចេកទេស

- 1 ពេលវិលីយ 2 រួមចាត់ពីខ្លួនគ្នាបាន • ពេលវិលីយ Age នឹងកំណត់ Birth_date
រាយវិន age នឹងកំណត់ខ្លះ

Stored vs. Derived Attributed

- Two or more attribute values are related
 - E.g., Birth_date vs. Age
 - 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.

ទីនេះ គឺ មិនអាច បង្កើតឡើង, ឬ មិនត្រូវ នៅក្នុង data base ដែលរួចរាល់ (facts, នេះ)

• NULL value (គឺជាដឹងទិន្នន័យ នៃ computer នេះតែប៉ុណ្ណោះ)

- Not applicable ឪនេះ
- Unknown មិនដឹង
 - The attribute value is missing or
 - Not known

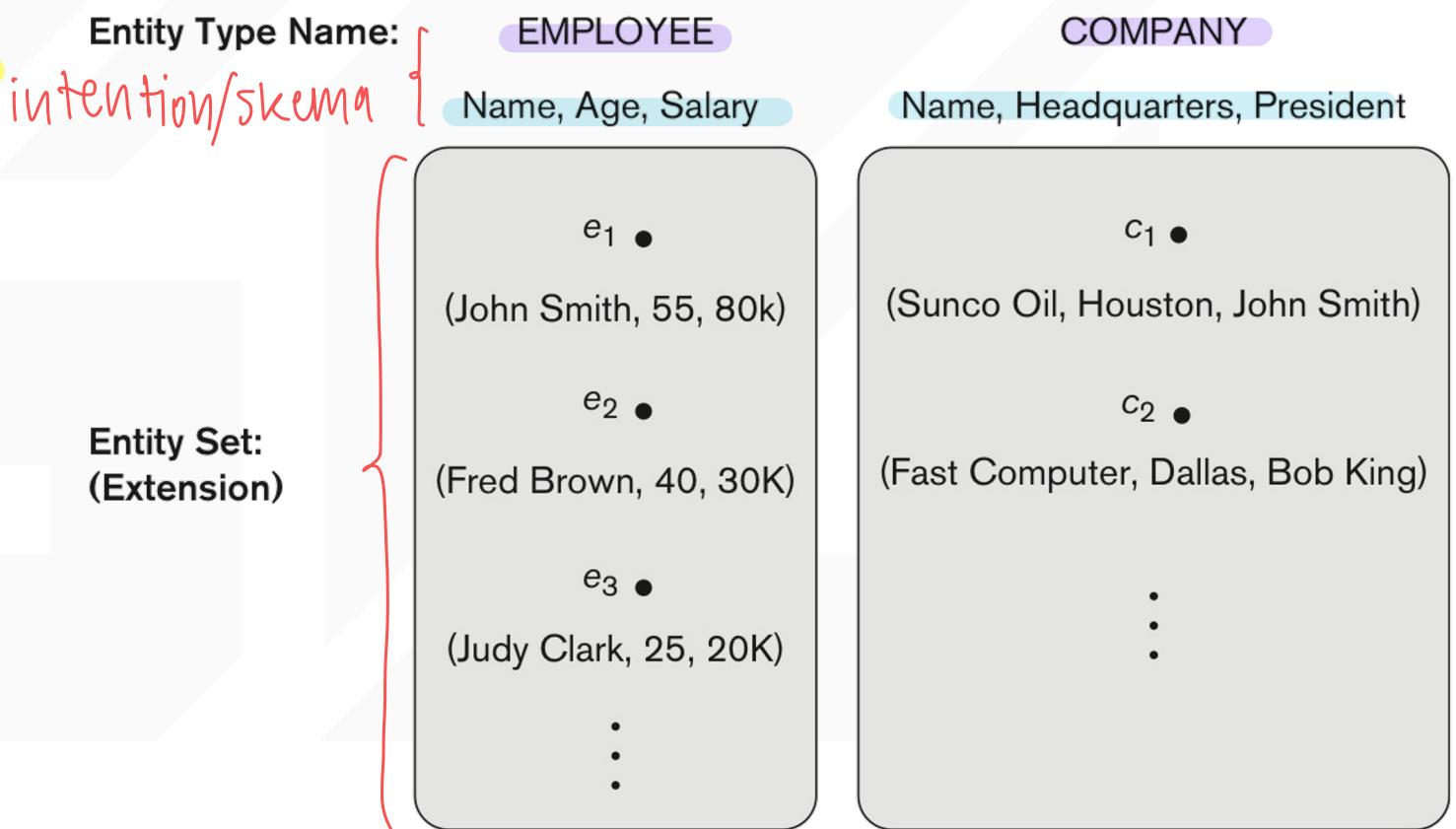


ទីនេះ - គឺសម្រស់តាមតំណែង
 - ឥឡូវក្រោម, ឬក្រោមថ្វី
 - វត្ថុខ្សោយក្នុងលេខទៅបើនិនិយត្រ

ទិន្នន័យ - A , NA ក្នុង, ឬនិយត្រ

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 (-entity ម៉ូល, ជំនាញ សេត ឬ Sets = nu.Set {})

- 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



재포

Key Attribute

- 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.
 - A key attribute may be composite.
 - VehicleTagNumber is a key of the CAR entity type with components (Number, State).

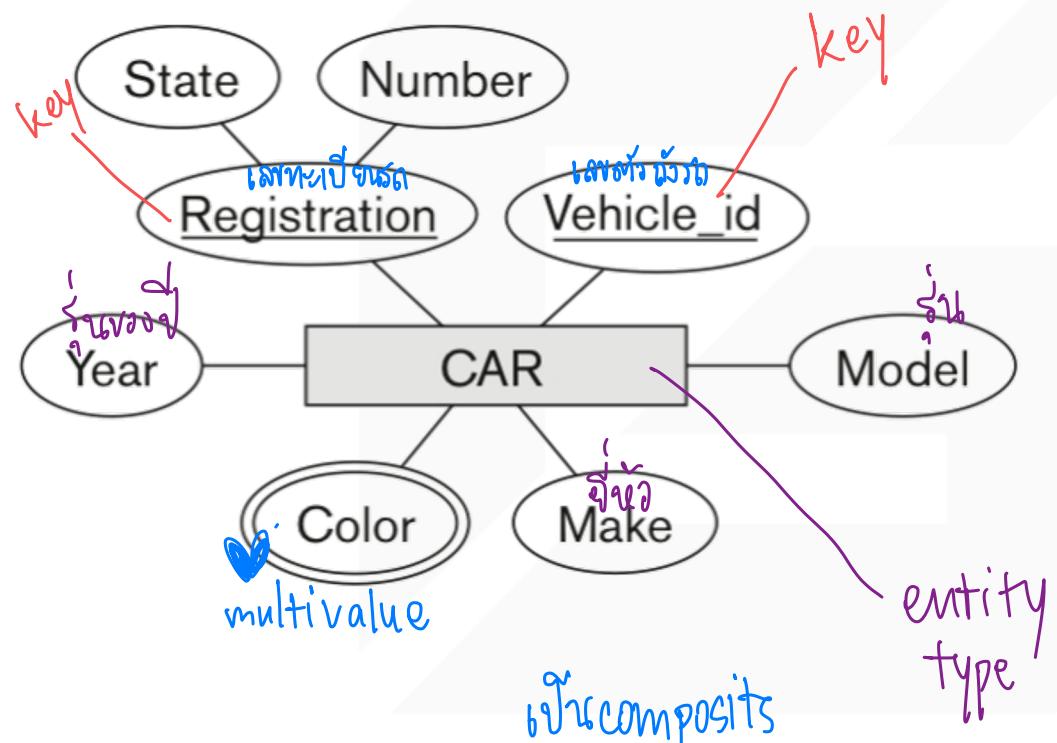
ចំណាំ នៅក្នុង
 • ឈ្មោះ - ឈាមសក្តី
 • លេខបញ្ជីរថយចន៍
 • តម្លៃសង្គកតែកម្រា
 • បេរិច្ឆេទសំណើនៅក្នុង

សំណើក្នុងព្រមទាំង នឹងត្រូវបានបង្កើតឡើង។
 គឺជាប្រភេទសំណើដែលត្រូវបានបង្កើតឡើង។

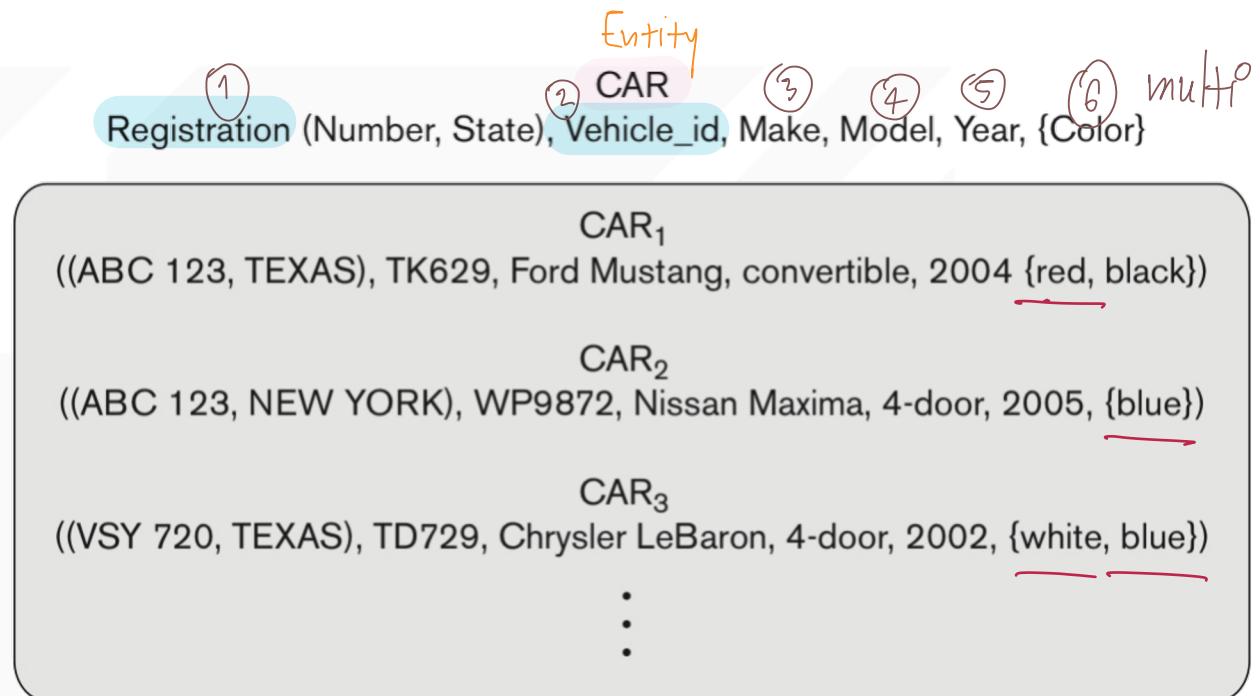


- An entity type may have more than one key.
 - The **CAR** entity type may have two keys:
 - **VehicleIdentificationNumber** (popularly called VIN) ตัวถัง
 - **VehicieTagNumber** (Number, State), aka license plate number ทะเบียนรถ
- Each key is underlined
 - Note: this is different from the relational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

ER Diagrams



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**



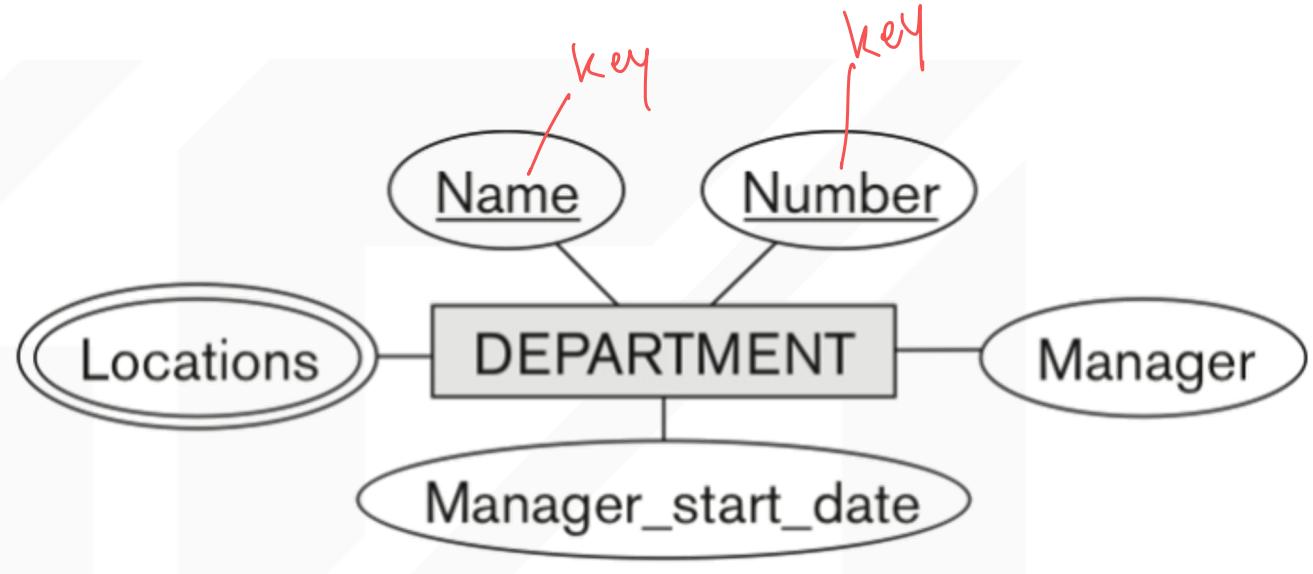
Figure 3.14
Summary of the notation for ER diagrams.

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	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

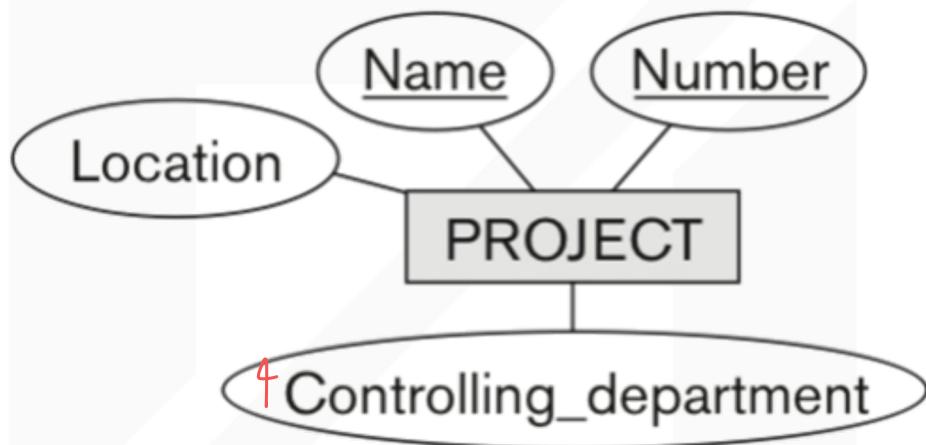
ជាគំសាយ

Initial Conceptual Design

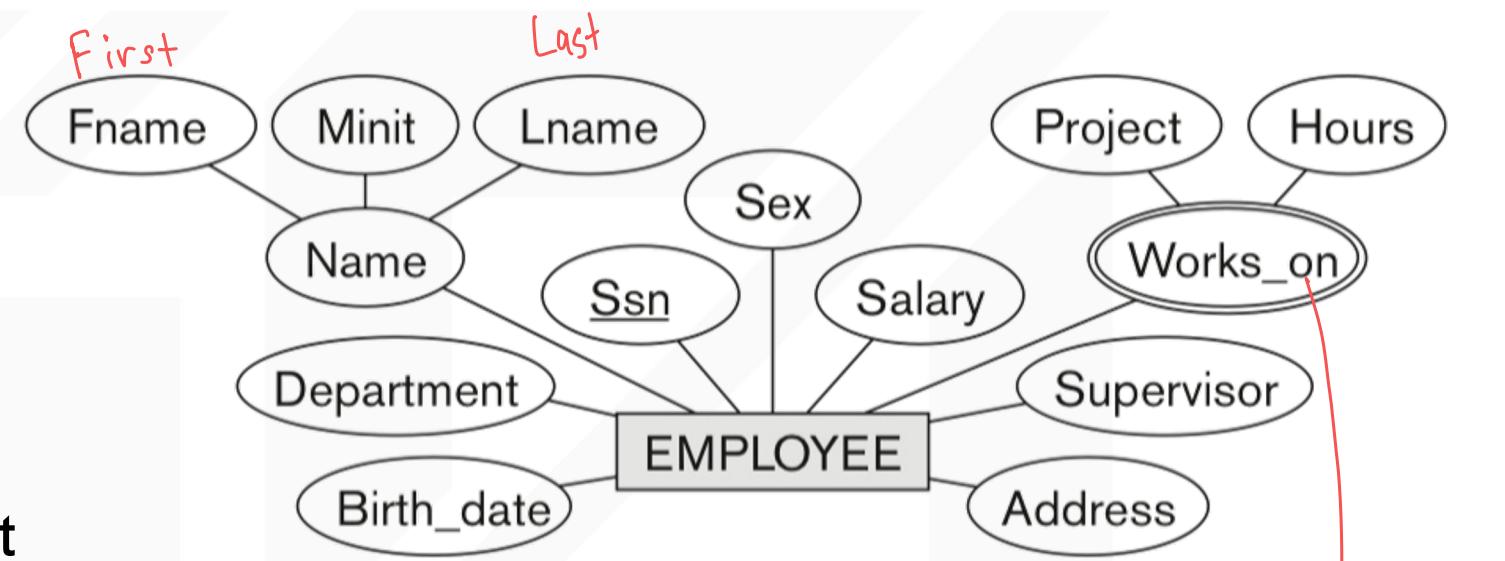
- The company is organized into **DEPARTMENTS**^(entity).
 Each department has a ^{attribute}**name**, ^{attribute}**number** and an employee who *manages* the department.
 We keep track of the ^{attribute}**start date** of the ^{attribute}**department manager**.
 A department may have ^{attribute}**several locations**.



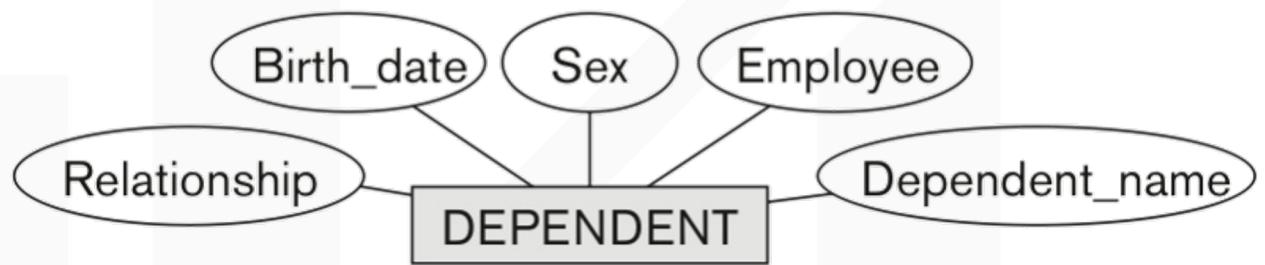
- Each department controls a number of PROJECTs
- Each project has a unique name, unique number and is located at a single location.

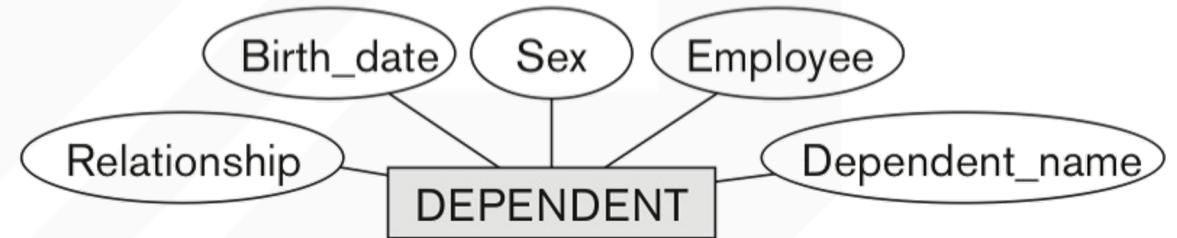
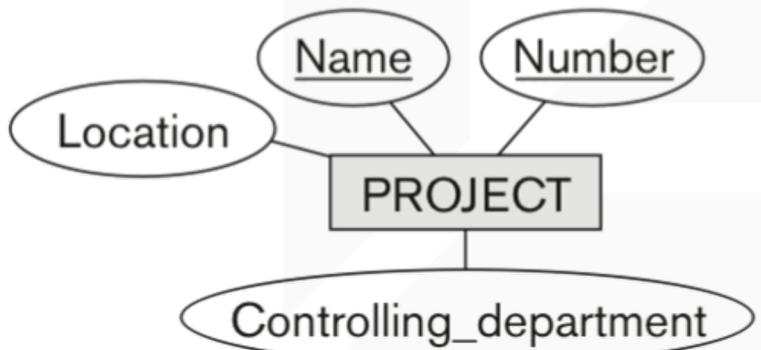
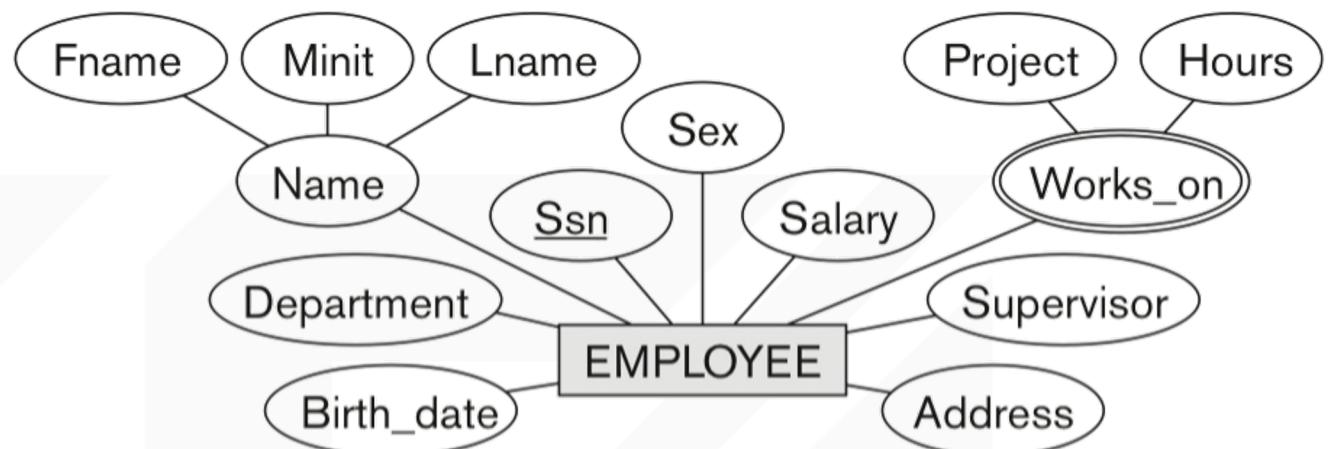
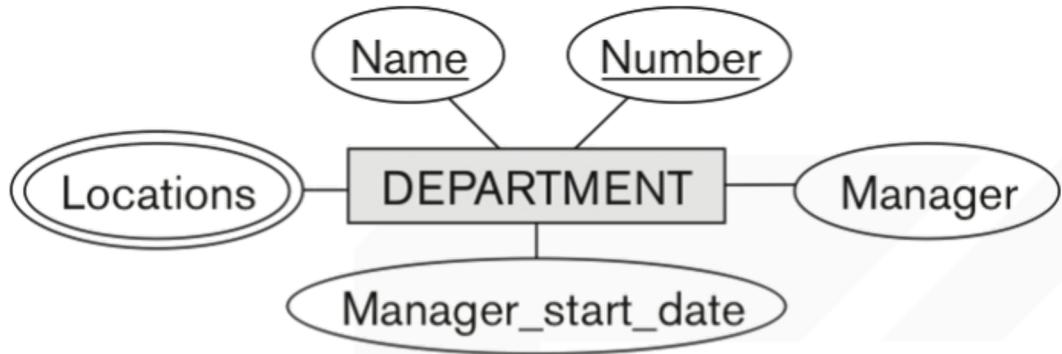


- The database will store each **entity** **EMPLOYEE's** social security number, address, salary, sex, and birthdate.
- Each employee works for one **department** but may work on several **projects**.
- The DB will keep track of the **number of hours per week** that an employee currently works on each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





- ER model has three main concepts:

- Entities (and their entity types and entity sets)
(ஆங்கிலம்)
- Attributes (simple, composite, multivalued) *இரண்டுபோல் ஒன்றாகவுள்ளது*
- Relationships (and their relationship types and relationship sets)

↪ அமல்காரன் (* முன் entities தாங்கி சொன், நீங் function வரே)

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, or
EMPLOYEE Franklin Wong manages the Research DEPARTMENT.

ມີຄວາມກົດໝາຍ ສິນເປົ້າ ນີ້ຈະອີງໃຫ້ໄລຍະສູງ

- 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

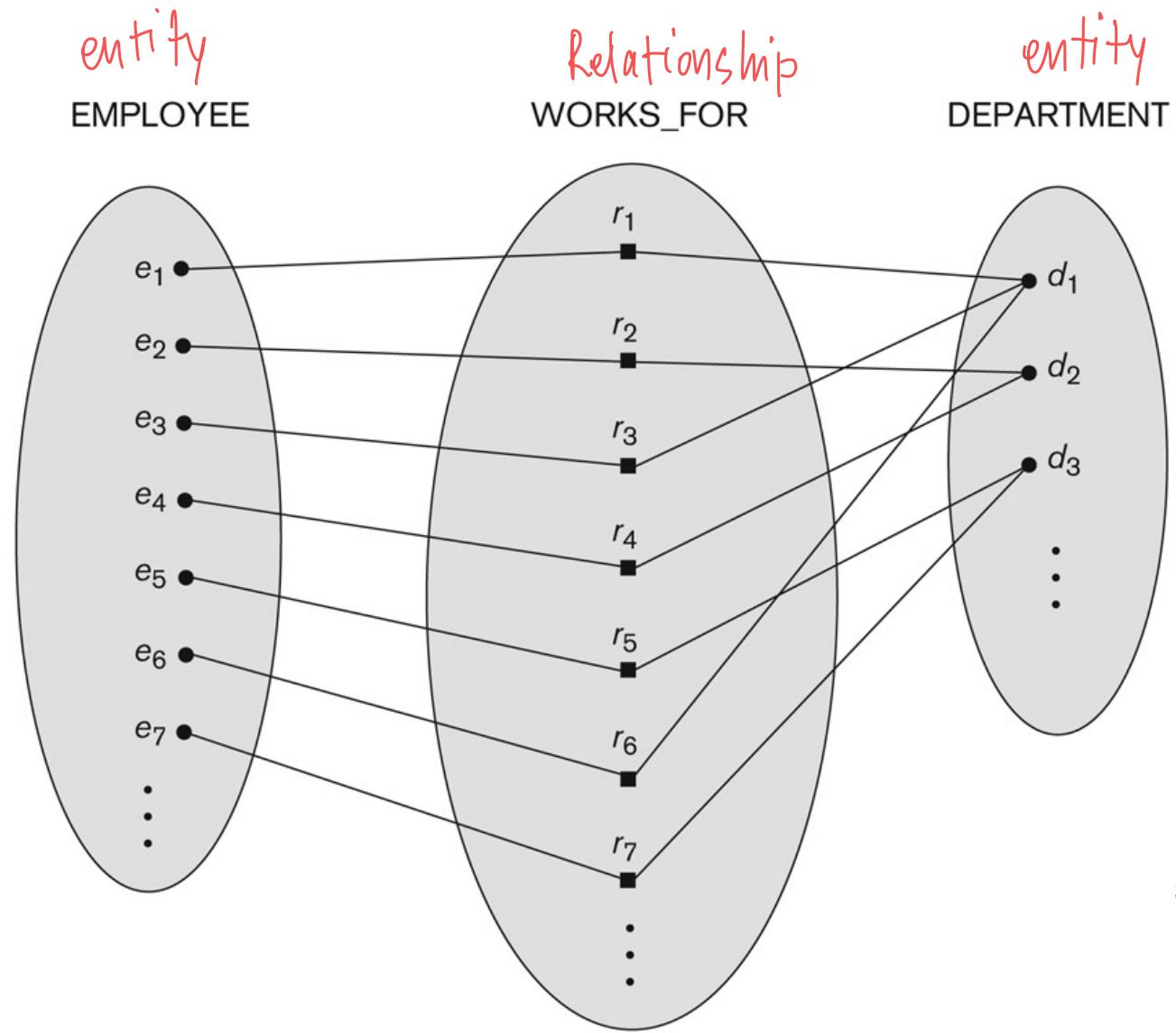


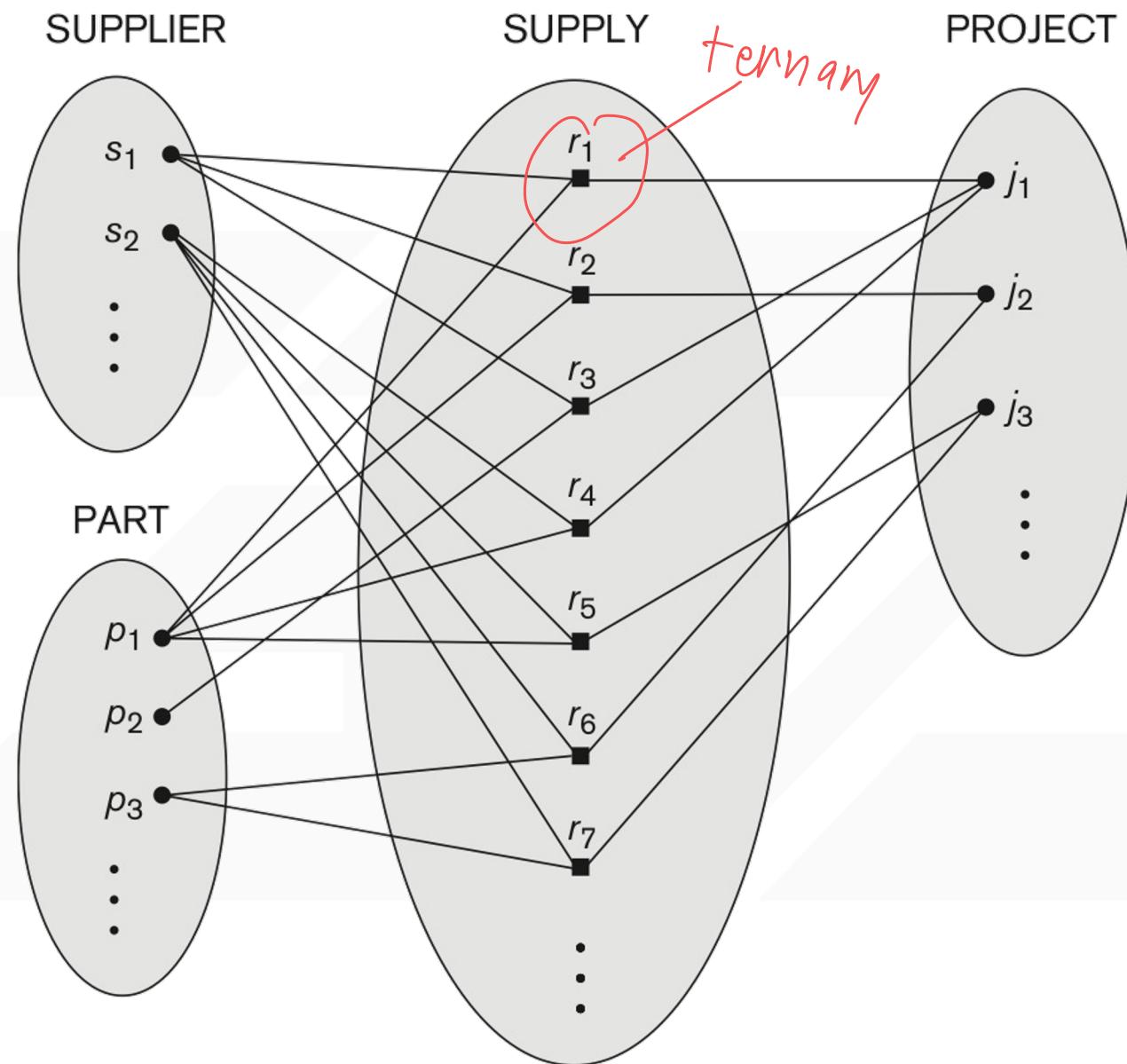
Figure 3.9
 Some instances in the WORKS_FOR relationship set, which represents a relationship type WORKS_FOR between EMPLOYEE and DEPARTMENT.

• 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.

4 . . < 4-ary
5 - . , 5-ary

binary & ternary



ຝັ້ງກັບຕົວເລີນ

• Recursive Relationship Type

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

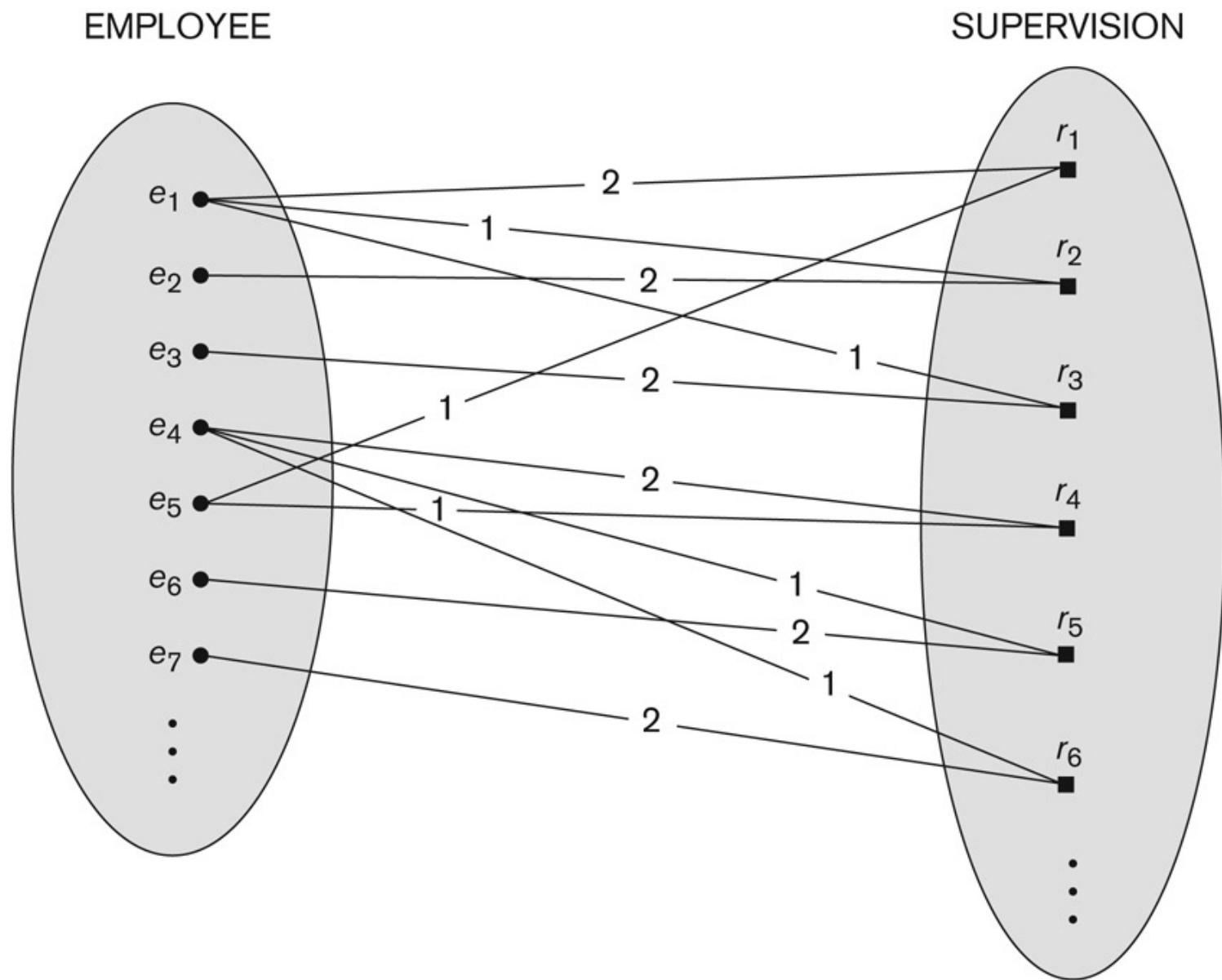
ກ້າງສັນຕິພາບ ສູນບະບາຫທີ່ມ່າງໄປຢູ່ກະເຄີນ



- Example: the **SUPERVISION** relationship

- EMPLOYEE ^{អ្នកចាយ ហោហេ} participates twice in two distinct roles:
 - supervisor (or boss) role ^{ផែកចាយ}
 - supervisee (or subordinate) role ^{រៀបចំ}
- Each relationship instance relates two distinct EMPLOYEE entities:
 - One employee in *supervisor* ^{boss} role ^{អ្នកចាយ ឬ អ្នករៀបចំ}
 - One employee in *supervisee* role

- 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).

**Figure 3.11**

A recursive relation-
ship **SUPERVISION**
between **EMPLOYEE**
(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: *Relationship*
 - 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

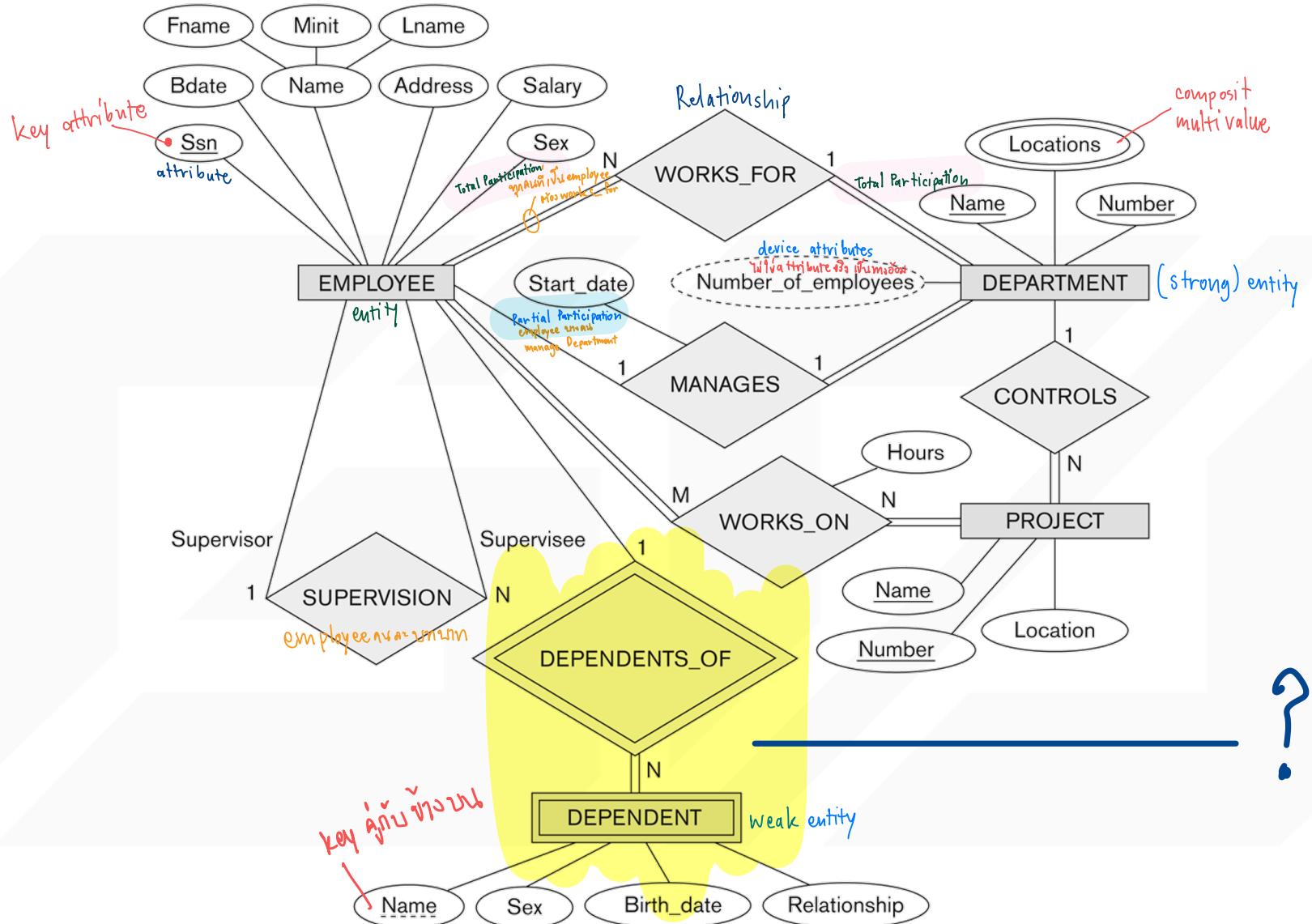


Figure 3.2

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

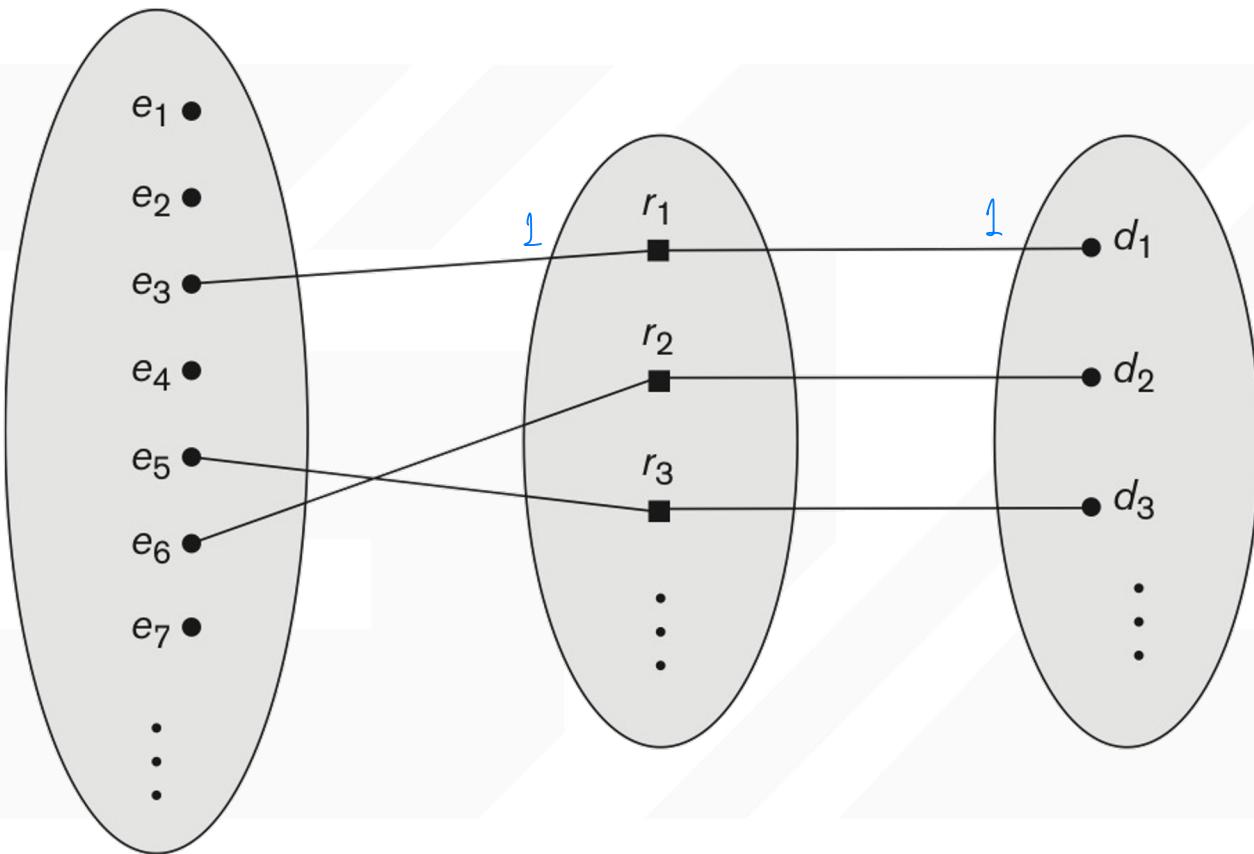
Constraints on Relationships

- Constraints on Relationship Types
 - Also known as **ratio constraints**
สัดส่วนของ.
 - **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)

EMPLOYEE

MANAGES

DEPARTMENT

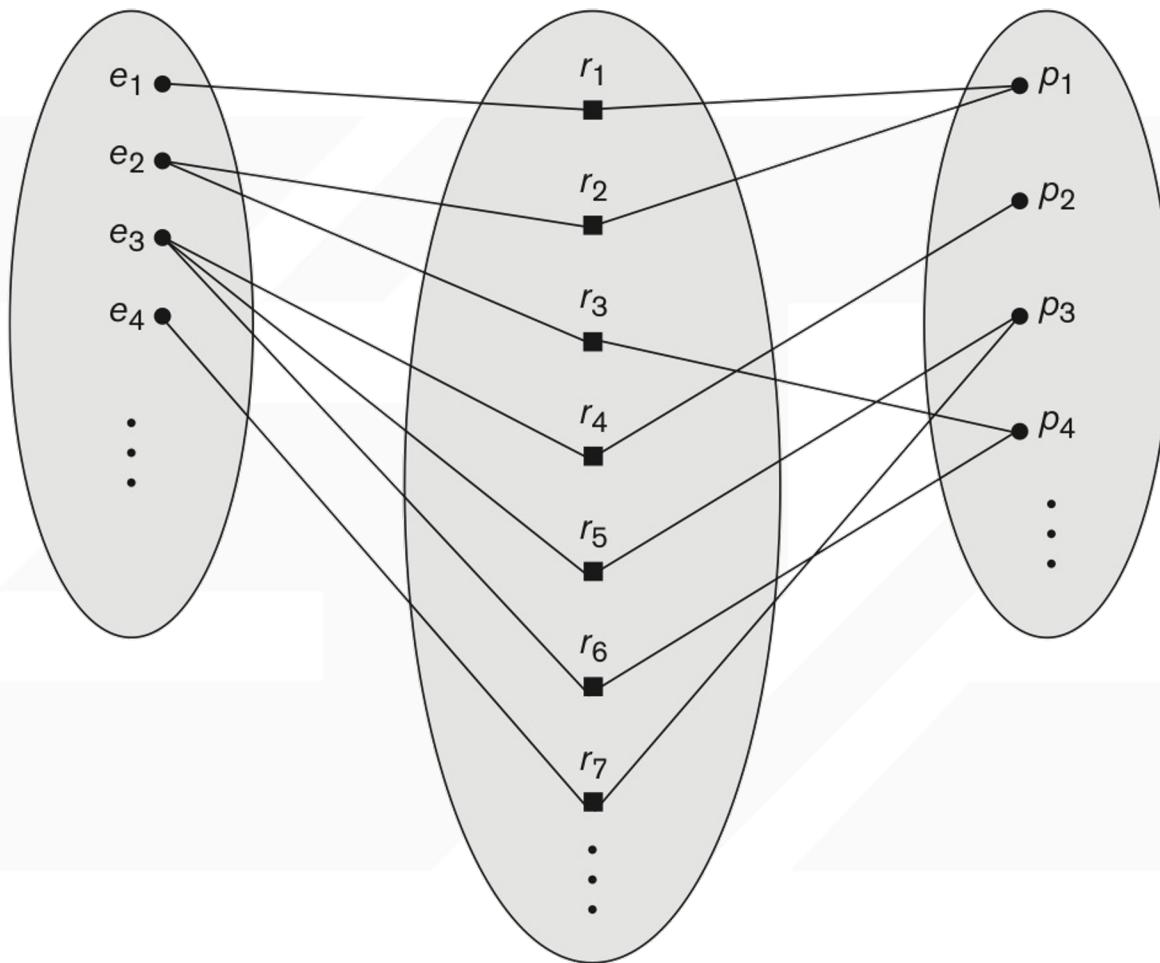


A 1:1 relationship, MANAGES

EMPLOYEE

WORKS_ON

PROJECT



An M:N relationship, WORKS_ON

Weak Entity Types

ໃຊ້ສັນ 2 ສັນ

ຄືດ entity ຂີ່ຕົລະອາຄີຍ ຊົ່ວໂມງ ຈາກທີ່ຈິນ ພາກໃໝ່ ແຕ່ລາວ ລວມສະບຸຽນ ເຊັ່ນ timestamp , Dependent

- 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**

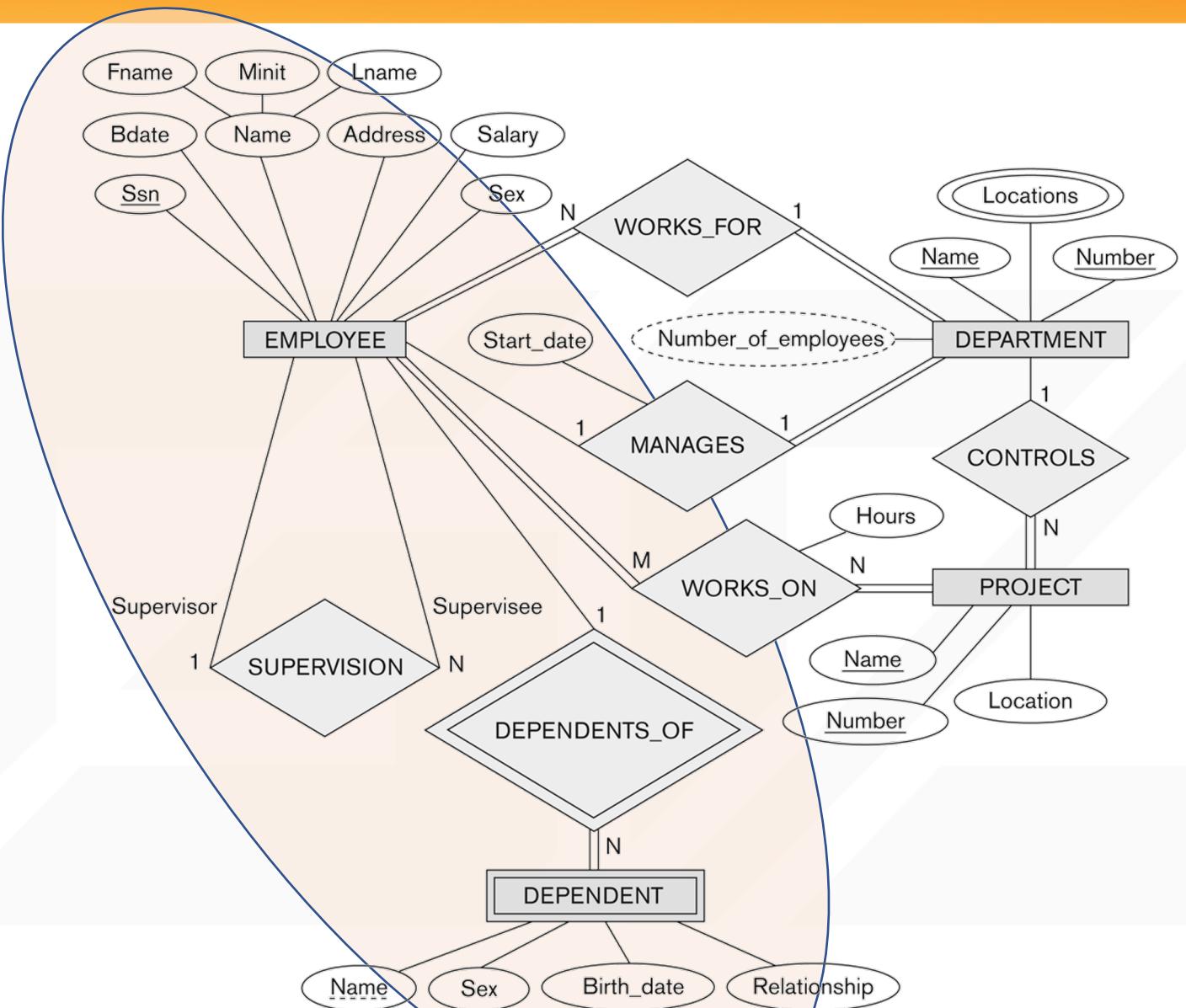


Figure 3.2

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

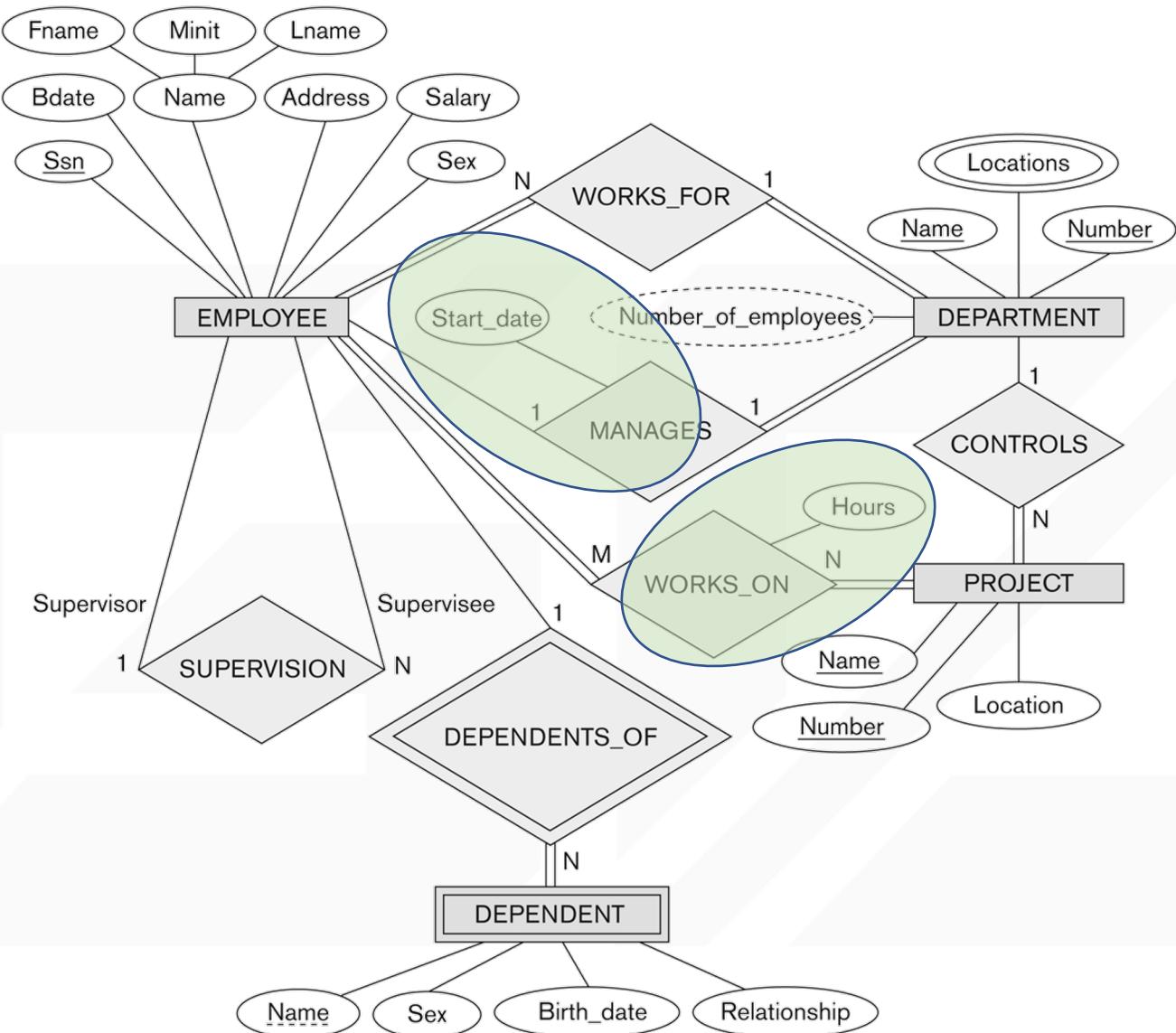
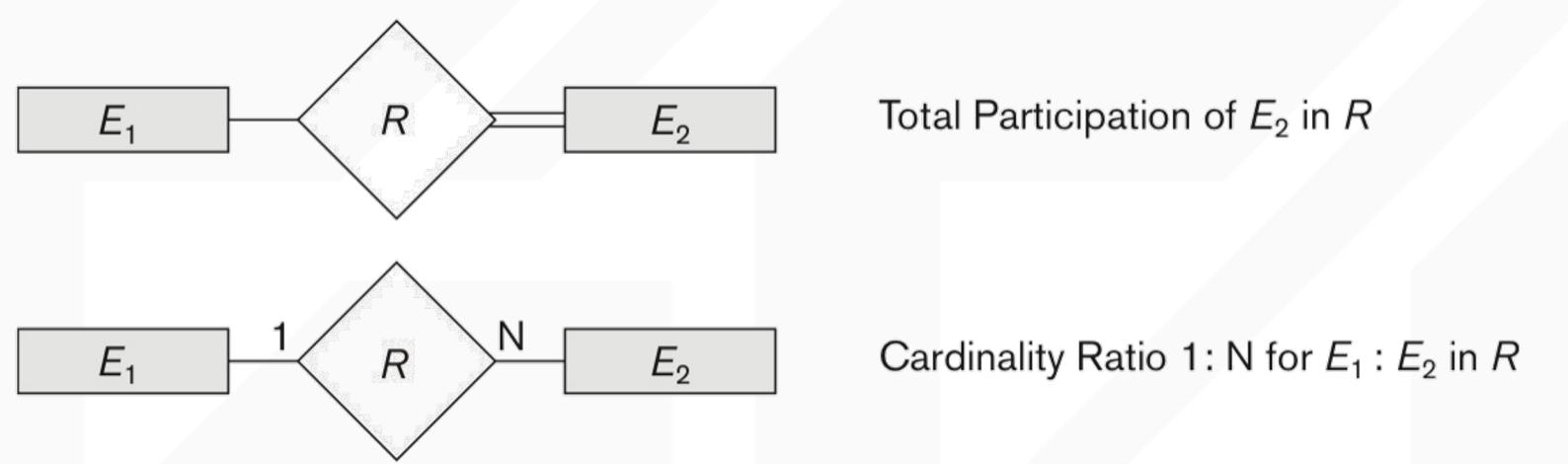


Figure 3.2

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.

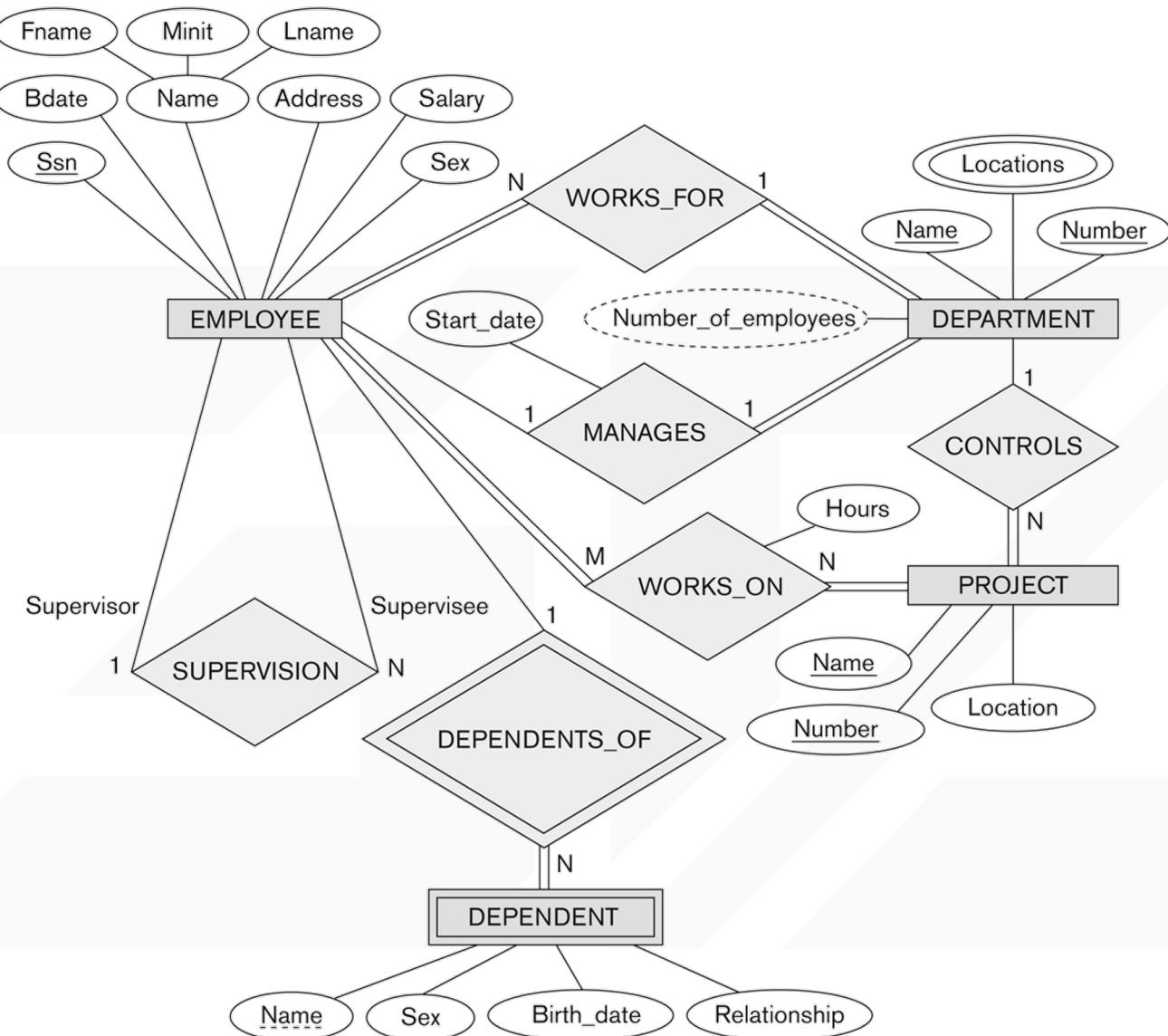


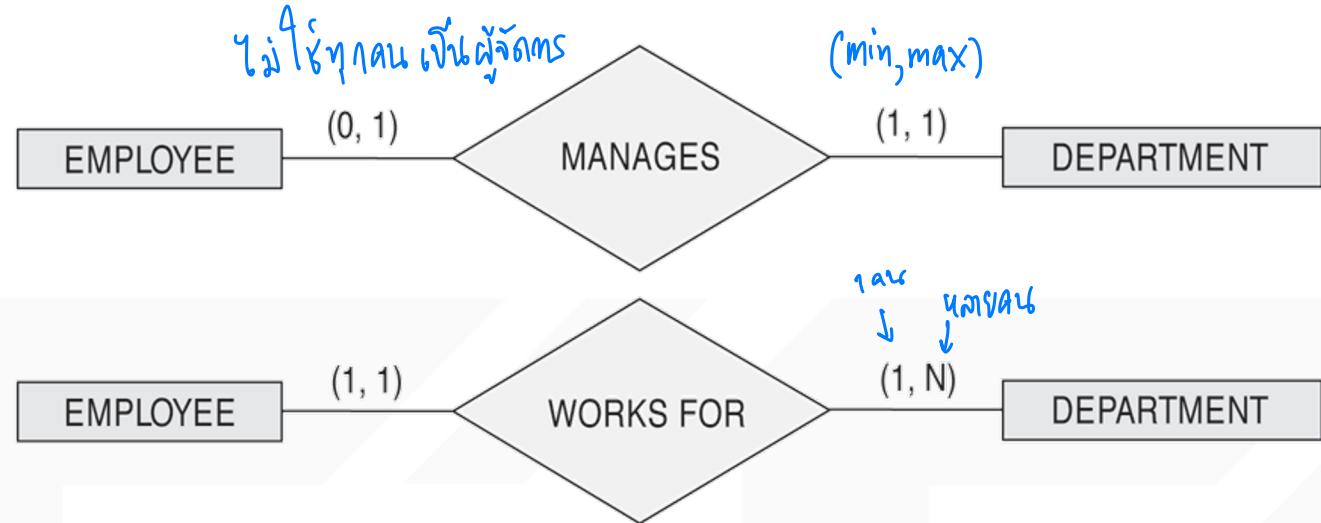
Figure 3.2

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 เป็น

- 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

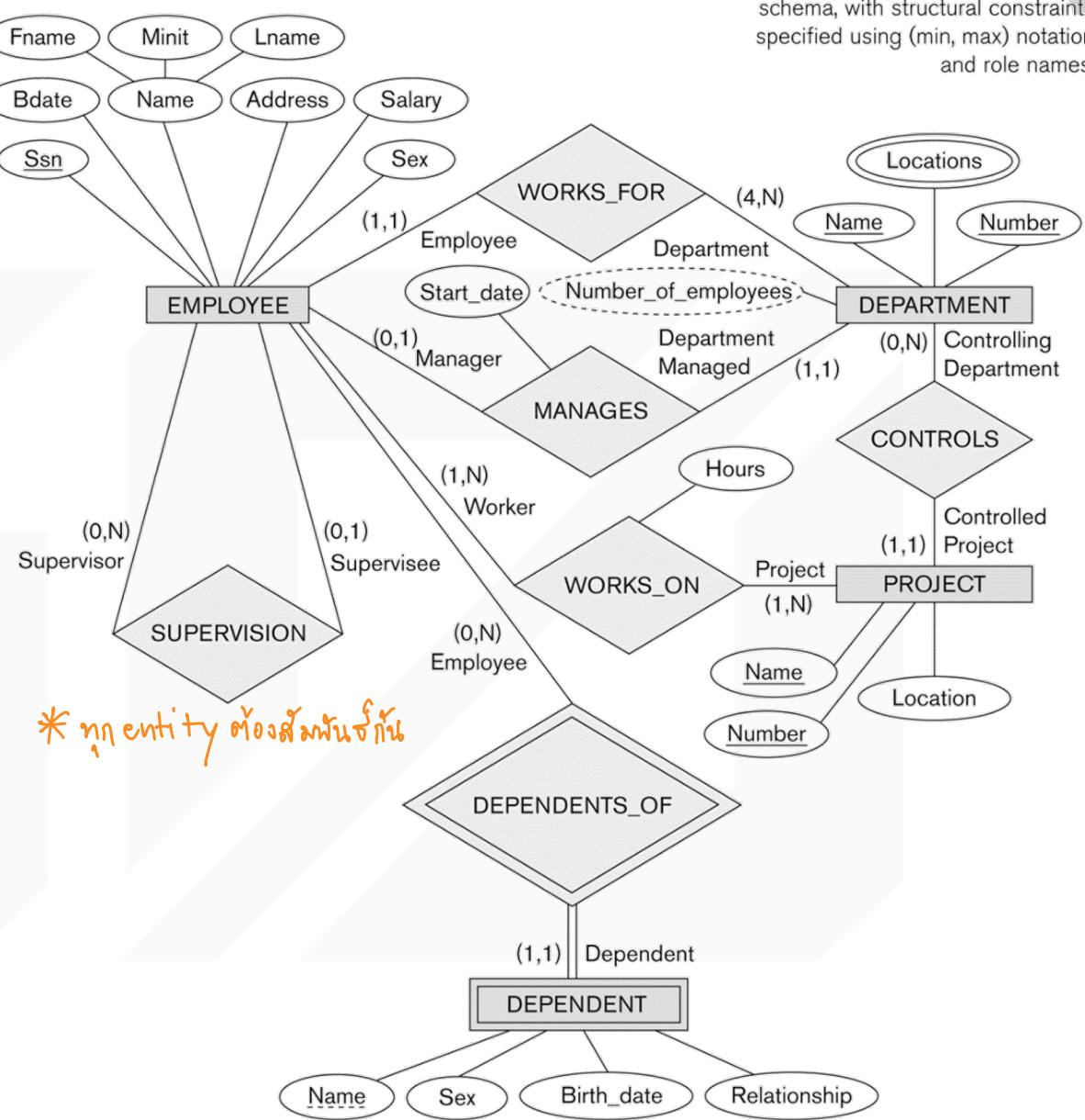
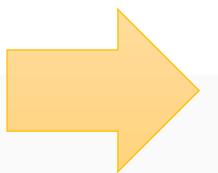
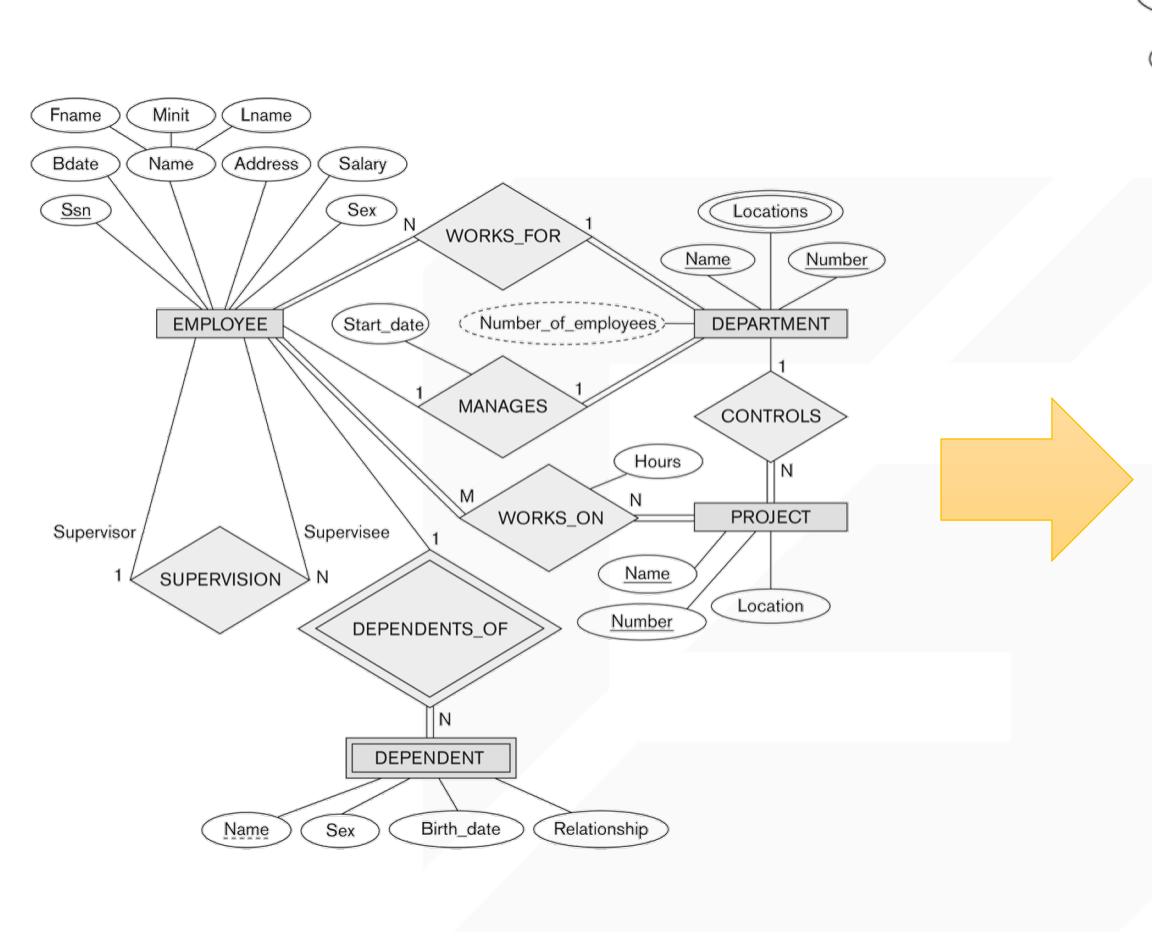


• Examples:

- 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

X ត្រូវការការពារ

សម្រាប់បង្កើតបណ្តុះបណ្តាល

