



Lecture4: Ch3. Data Modeling

Using the Entity–Relationship (ER) Model

part1

Dr. Alaa eldin Abdallah Yassin

Outline

- ER Model Concepts
- Types of Attributes.
- Entity Types and Entity Sets.
- ER Diagram
- Constraints on relationships
 - Relationships and Relationship types
 - Existence Dependency
- Alternative (min max) notation
- ER diagram for Company database

ER Model* Concepts

► Entities and Attributes

- **Entities***: are specific persons, places, events, **objects** in the **user** environment about which the user **needs to keep data**.

- ❖ For example : EMPLOYEE, office, task,...

- **Attributes***: are **properties used to describe entity**.

- ❖ For example : EMPLOYEE entity may have the attributes Name, SSN, Address, Sex, BirthDate,...

- ❖ **Relationships**: a relationship **relates two or more entities with a specific meaning**.

- ❖ For example : EMPLOYEE *Ahmad* **Works on** ProductX PROJECT.

- ❖ EMPLOYEE *Hassan* **Manages** the Research DEPARTMENT

Types of Attributes

- **Composite** versus **Simple** (Atomic) Attributes.
- **Single-Valued** versus **Multivalued** Attributes.
- **Stored** versus **Derived** Attributes

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(attributes).

- For example :

- ❖ **Name** (FirstName, MiddleName, LastName).

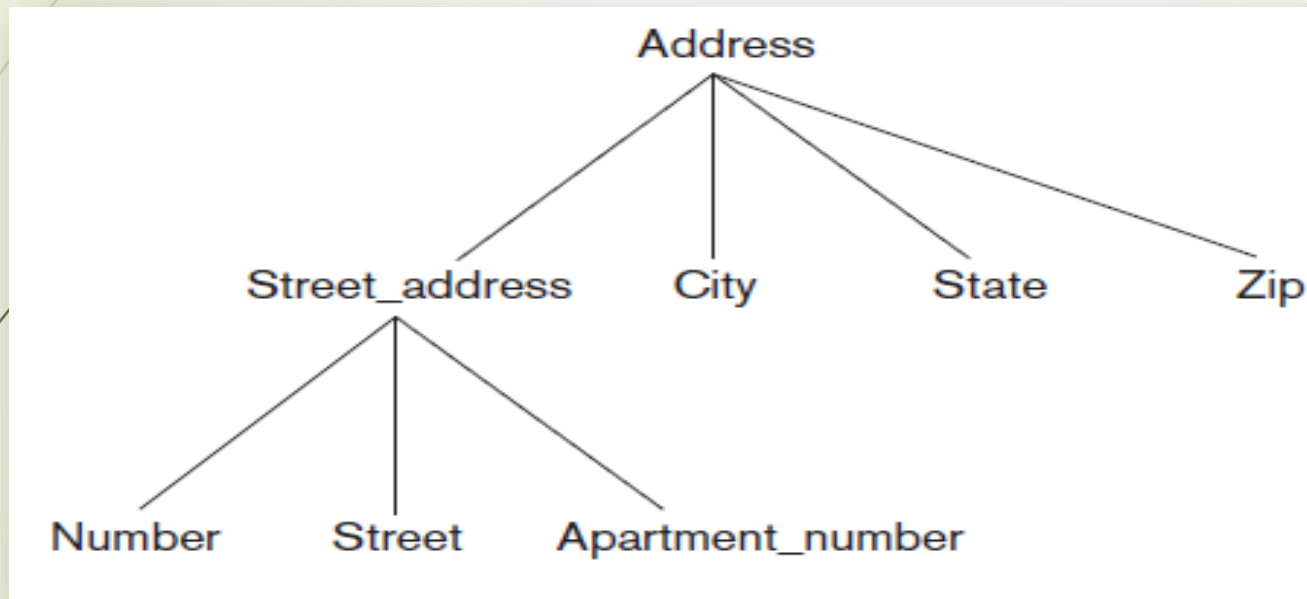
- ❖ **Address** (House#, Street, City, state, Country..)

➤ Multi-Valued

- An Entity may have a multiple values for the attributes.

- ❖ For example : Color of a CAR {black, white}.

- ❖ Qualifications { BSC, MSC, PhD,...}



← composite attributes

Types of Attributes...

➤ **Stored Attributes***

➤ Attribute that it take its value from user.

❖ For example : Salary, Bonus

❖ BirthDate

➤ **Derived Attributes***

❖ For example : $\text{NetSalary} = \text{Salary} + \text{Bonus}$

❖ Age : I can calculate from BirthDate attribute

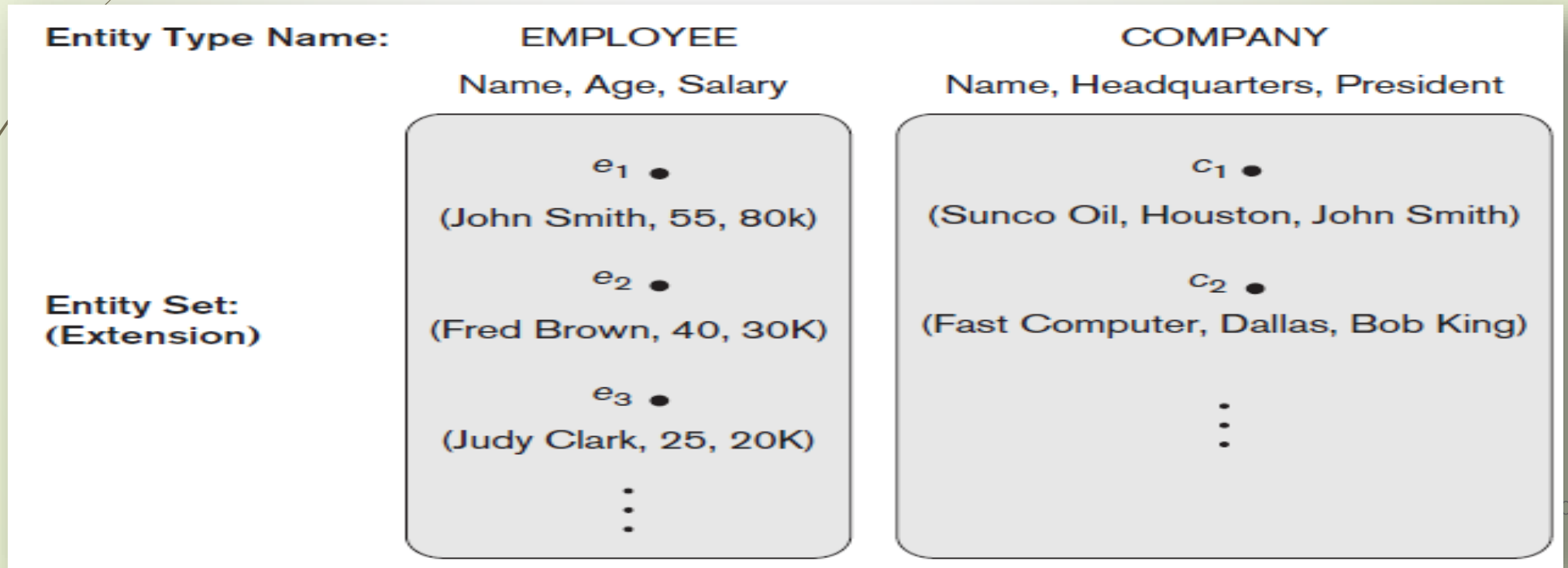
❑ **Complex Attributes**

❖ It's a **multi-valued** attribute and **composite** attribute.

❖ For example : $\text{Phone} = (\text{Countcode}, \text{govCode}, \text{Phone\#}, \text{Phone\#})$


Entity Types and Entity Sets

- **entity type** defines a “ *collection (or set) of **entities** that have the same **attributes***”.
- Each **entity type** in the database is **described** by its **name** and **attributes**.



ER Diagram

- **An entity type** is represented in ER diagrams as a **rectangular** box enclosing the entity type name. 

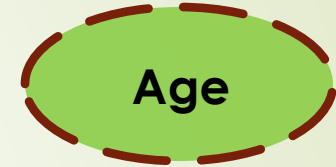
- **Attribute** names are enclosed in **ovals** and are attached to their entity type by straight lines. 

- **Composite attributes** are attached to their component attributes by straight lines. 

- **Multivalued attributes** are displayed in **double ovals**. 

ER Diagram...

➤ derived attributes are displayed in dotted ovals



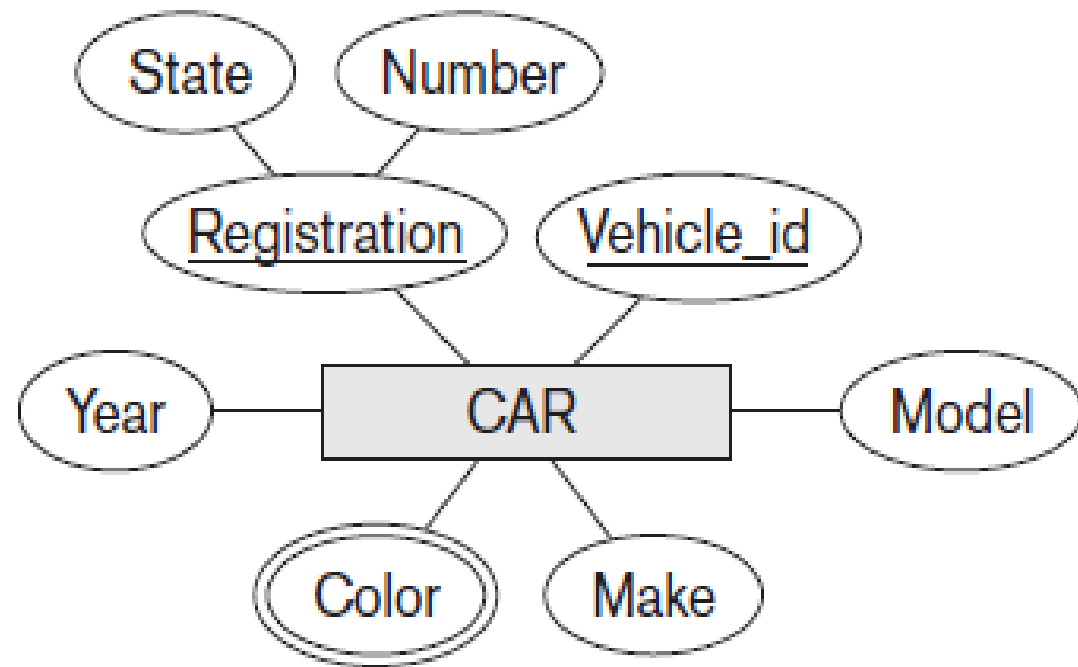
➤ **Key Attributes** of an Entity Type:

- ❖ An entity type usually has one or more attributes whose values are **distinct (uniqueness)** for each individual entity in the entity set.
- ❖ Such an attribute is called a key attribute, and its values can be used to identify each entity uniquely.
 - For example, the **Name** attribute is a key of the **COMPANY** entity type because no two companies are allowed to have the same name.
 - For the **PERSON** entity type, a typical key attribute is **SSN** (Social Security number).
- ❖ Sometimes **several attributes together form a key**, meaning that the combination of the attribute values must be distinct for each entity.
- ❖ each **key attribute** has its name underlined inside the oval



Ex: entity type with key attributes

(a)



The **CAR** entity type with two key attributes, Registration and Vehicle_id
ER diagram notation.

Ex: entity type with key attributes...

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})

⋮

CAR

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

Entity set with three entities.

➡ True or False

1. The same Entity may have more than one key attributes [T/F] **T**
2. The entity may have no any key attribute [T/F] **T : weak entity***

➡ **Weak entity** : represented by **double rectangle**.

dependent

➡ **partial key** : A weak entity type normally has a **partial key**, which is the attribute that can uniquely identify weak entities that are *related to the same owner entity*. (partial key attribute is underlined with a dashed or **dotted line**)

- ❖ In our example, if we assume that no two dependents of the same employee ever have the same first name, the attribute **Name** of DEPENDENT is the partial key.
- ❖ In the worst case, a composite attribute of all the weak entity's attributes will be the partial key.

Constraints on Relationships

► Constraints on **Relationship types**

❖ Also known as **ratio constraints**.

❖ **Cardinality Ratio** (specific maximum participation)

1. One-to-One (1:1)
2. One-to-Many (1:N) or Many-to-One (N:1)
3. Many-to-Many (M:N)

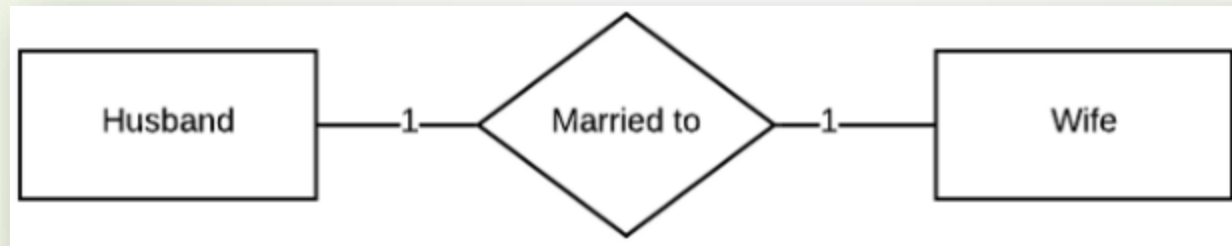
❖ **Existence Dependency** Constraints (specific minimum participation)

❖ Also called **participation constraint**.

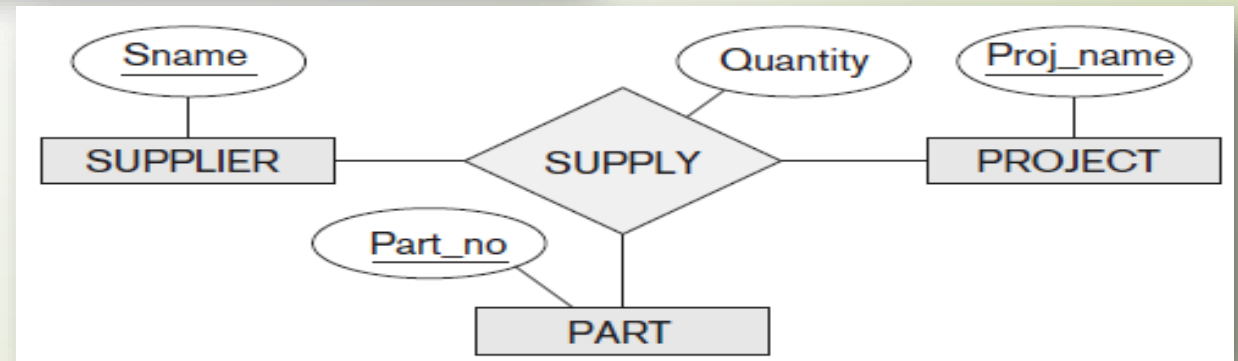
1. Zero (optional participation, not existence-participation)
2. One or more (mandatory participation, existence-participation)

Relationships and Relationship types

- The **degree** of relationship type is the **number of participating entity type**.
- Relationship represented by **Diamond** 
- 1. Both MANAGES and WORKS_ON are **binary** relationship.



2. Ternary relationship

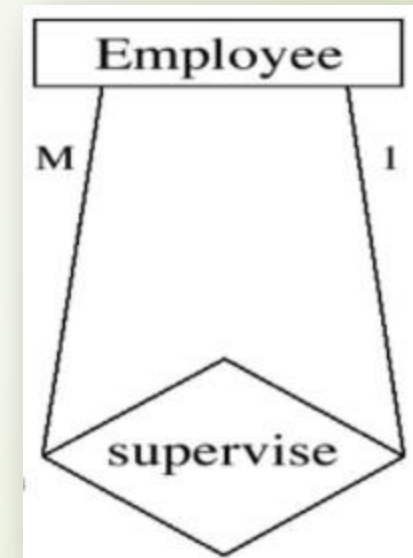


Relationships and Relationship types...

➤ Recursive relationship

➤ Entity related to itself.

❖ For example : **EMPLOYEE** supervise another **EMPLOYEEs**



Cardinality* of Relationship

1. One-to-One

- Each Entity in the relationship will have exactly one related entity. (**1:1**)

❖ Ex: EMPLOYEE and DEPARTMENT (manage relationship)*

2. One-to-Many

- an Entity on one side of the relationship can have many related entities, but an entity on the other side have a maximum of one related entity. (**1:M**)

❖ Ex: DEPARTMENT and PRPJECT (manage relationship)*

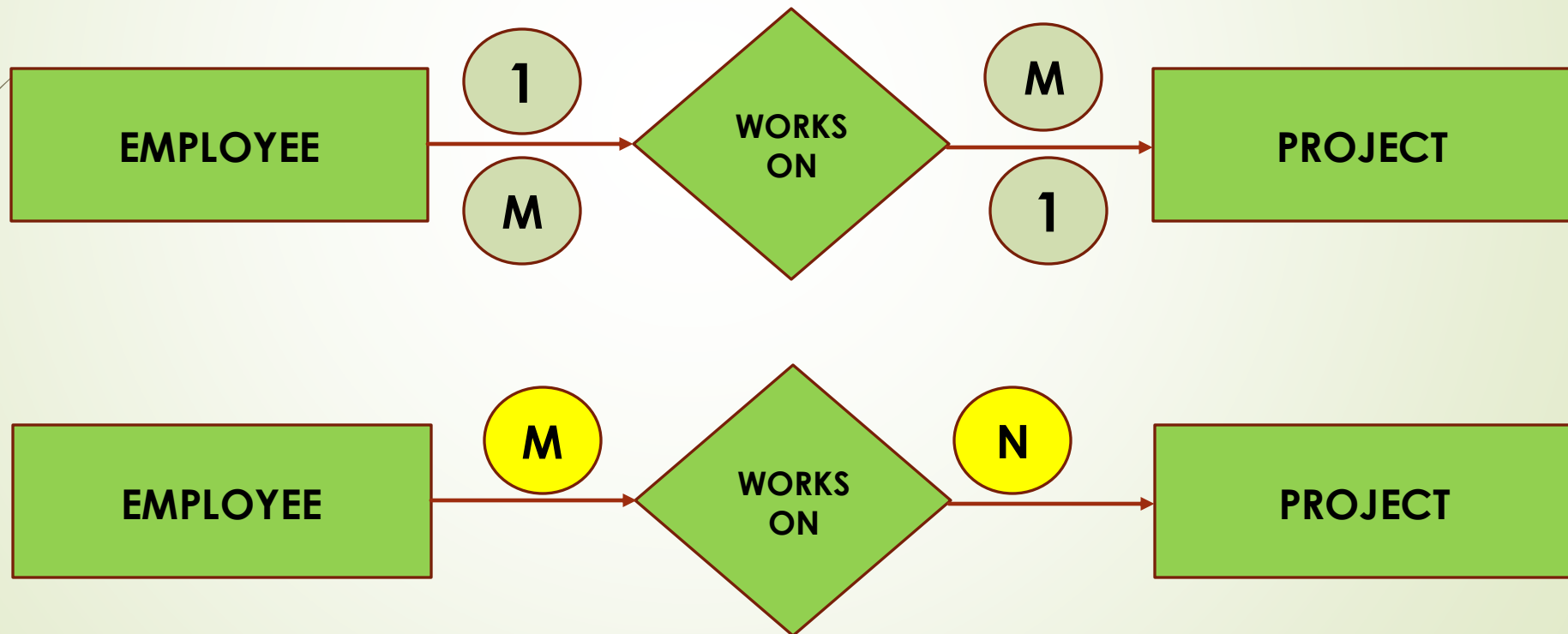
3. Many-to-Many (**M:M**)

- Entities on both side of the relationship have many related entities from the other side.

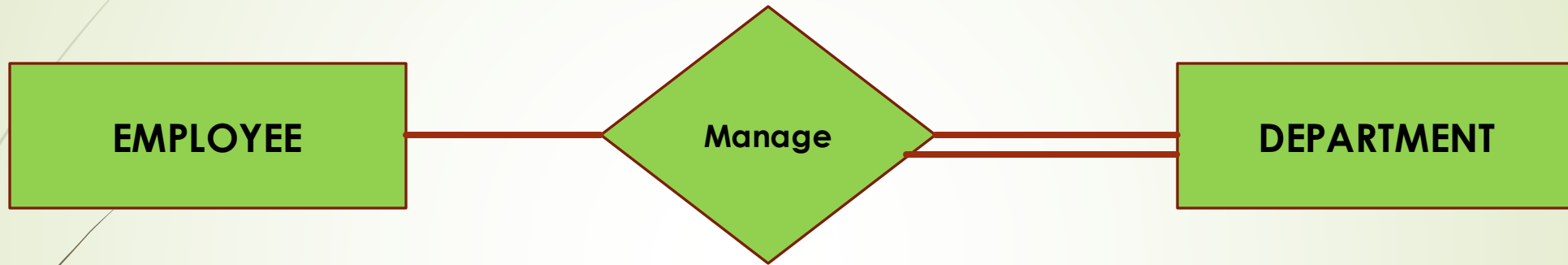
❖ Ex: EMPLOYEE and PRPJECT (manage relationship)*

Cardinality of Relationship...

many to many

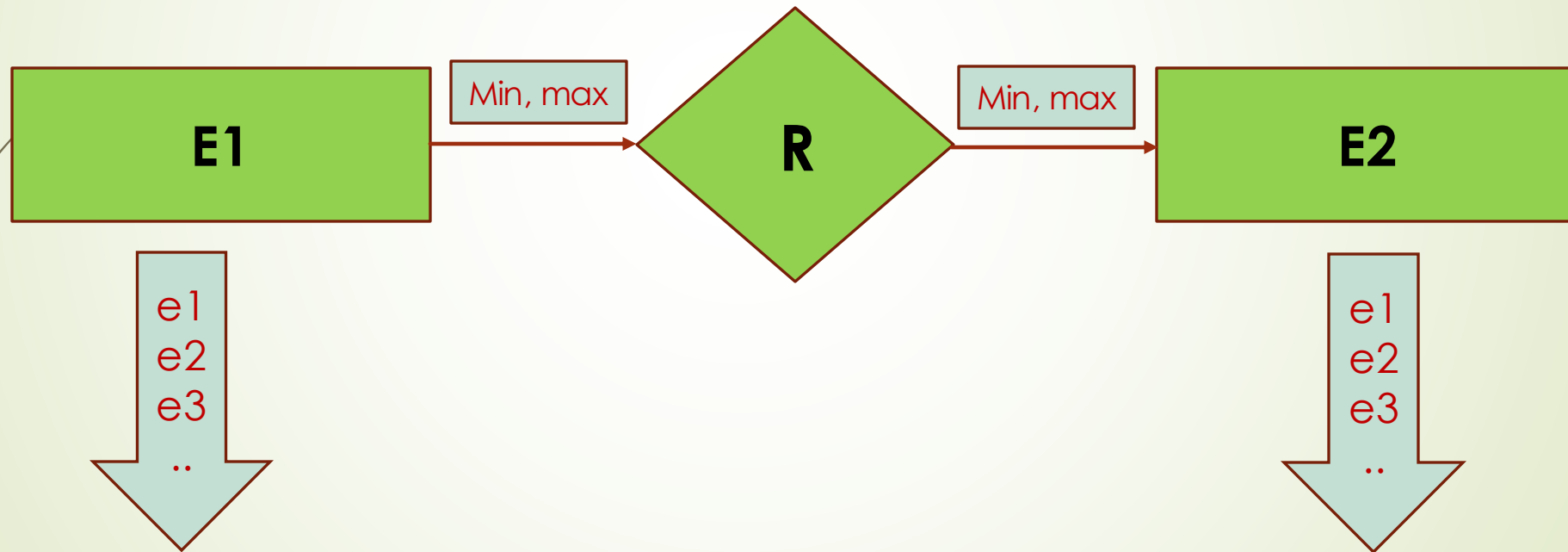


Total & partial participation Relationship



- Not all EMPLOYEE manage DEPARTMENT (**partial participation**)
- But all DEPARTMENTS should manage by EMPLOYEE (**total participation**)

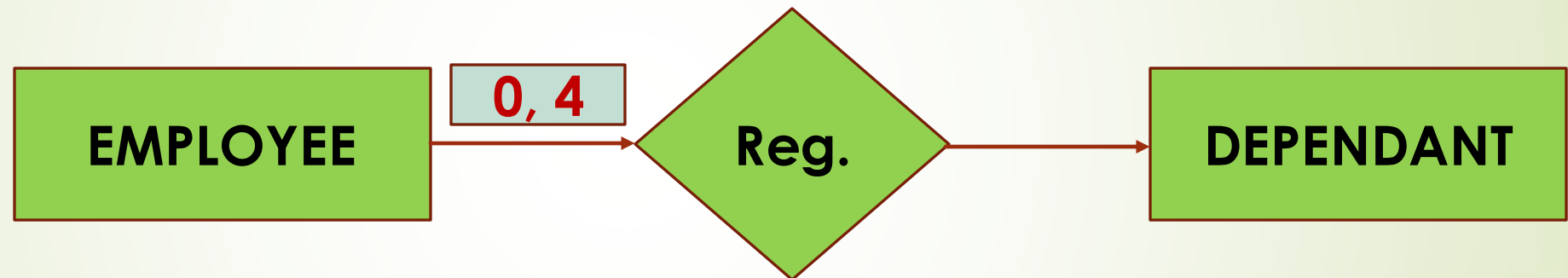
Alternative (min max*) notation* for Relationship structural constraints



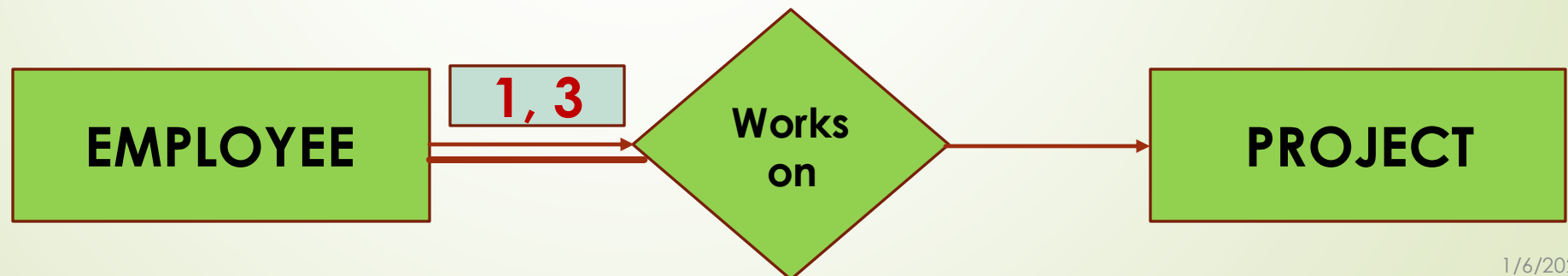
- Ex: The employee has the right to register **4** members of his family in the health care service (**max = 4**)

Total & partial participation Relationship...

- Ex1: The employee has the right to register **4** members of his family in the health care service ($\text{max} = 4$)



- Ex2: The employee must participate in at least one project and at most 3 projects ($\text{min} = 1$ & $\text{max} = 3$)



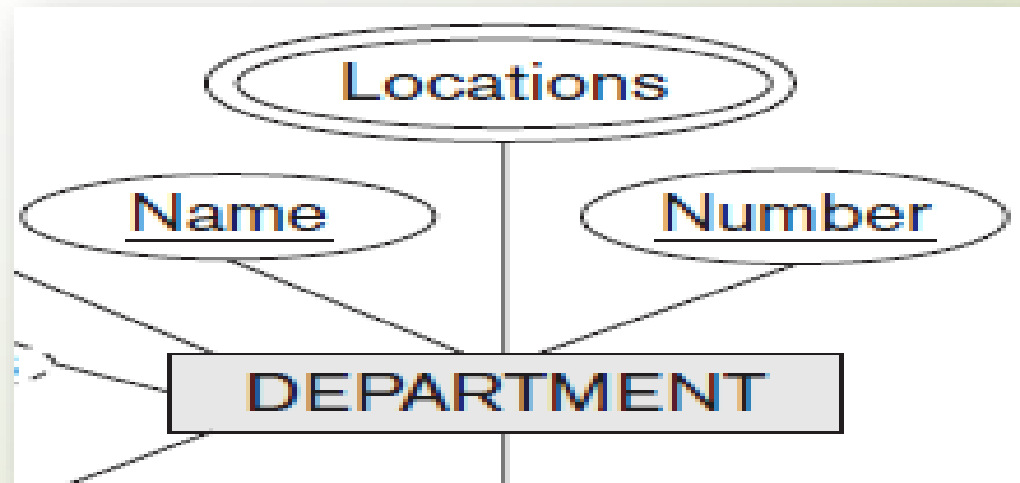
- Note** : if we found double line in relation (total partial); by default, the $\text{min} = 1$

ER Design for the **COMPANY** Database



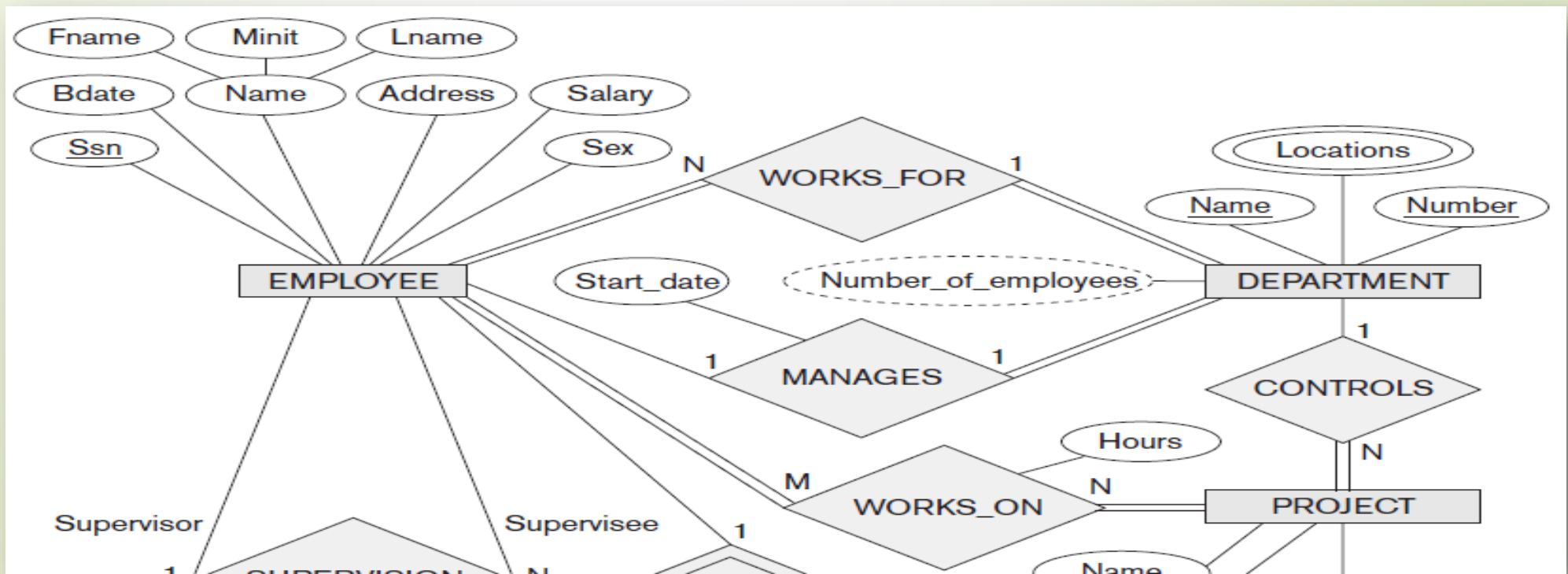
Example of **Company** database **DEPARTMENT** entity

- ▶ We need to create database schema design based on the following simplified **requirements** of the **COMPANY** database:
 - ❖ The company* is organized into **DPARTMENT**s.
 - ❖ Each **department** has a unique name*, unique number and an employee who *manages* the department.



Example of **Company** database **DEPARTMENT** entity...

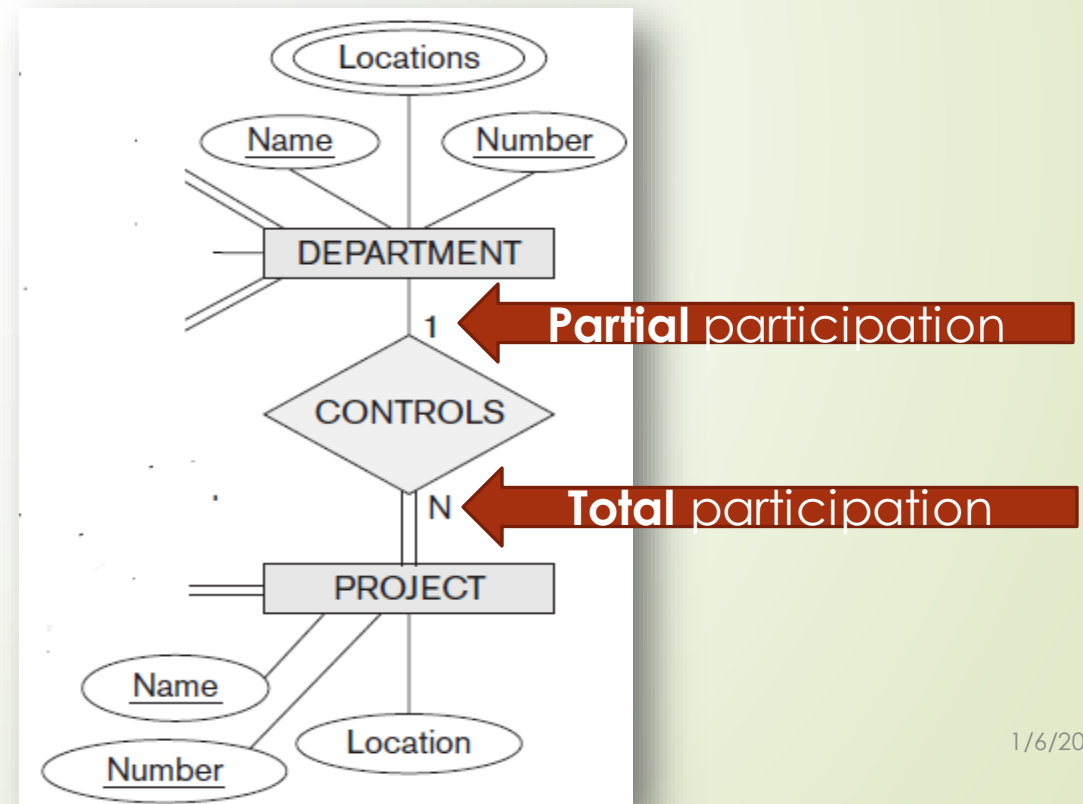
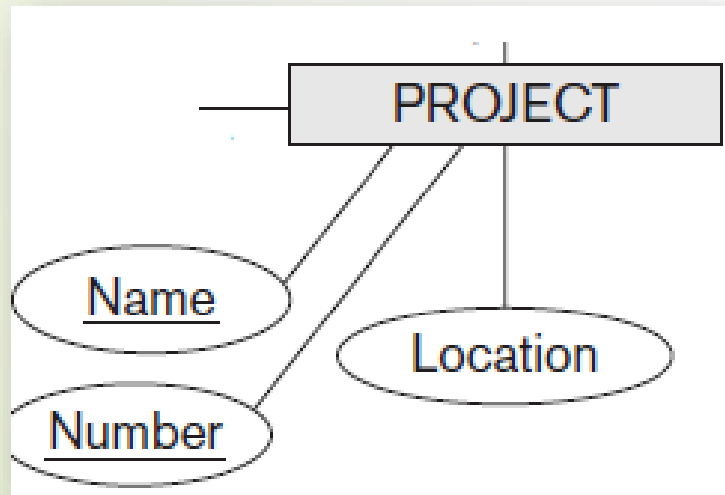
- ❖ Each **department** has a unique name*, unique number and an employee who *manages* the department. We keep track of the start date of the department manager. A department may have several locations*.



Example of **Company** database...

PROJECT entity

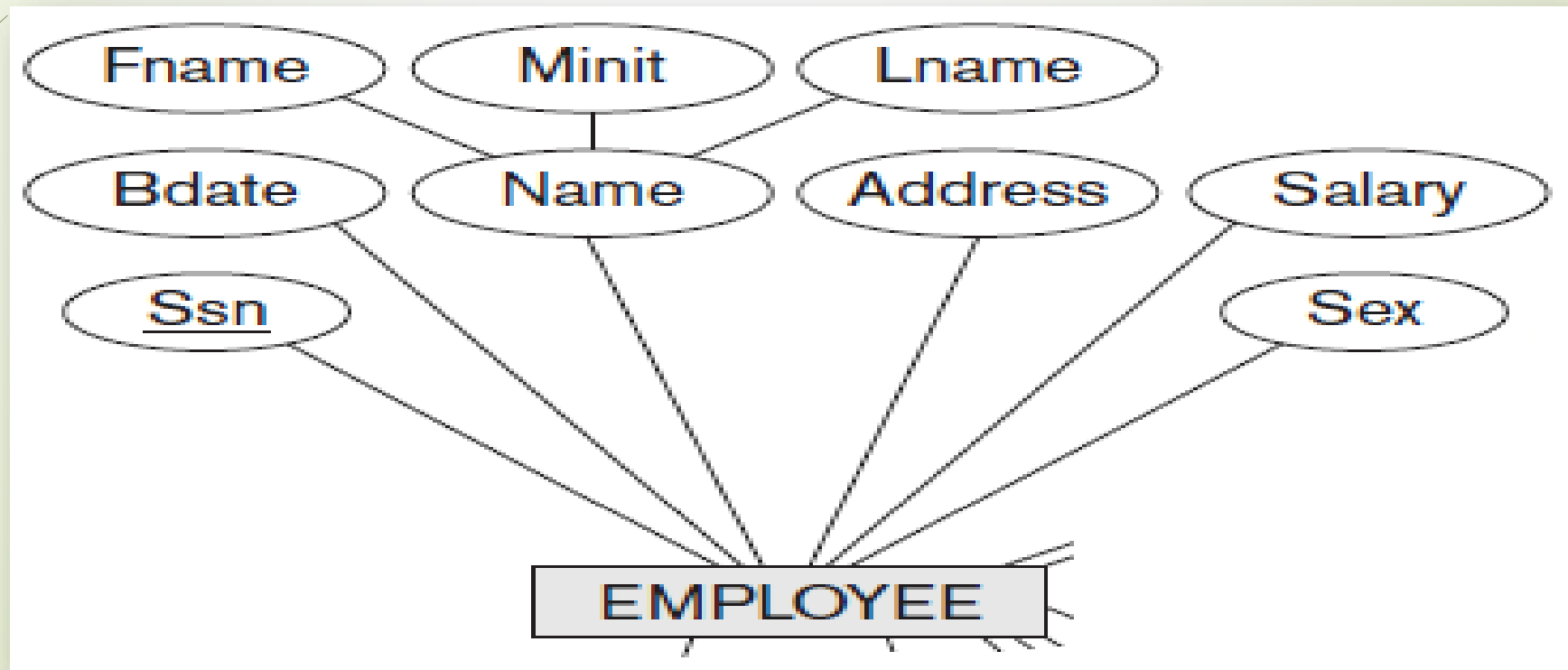
- ❖ Each **department** *controls* a number of **PROJECTS**. Each project has a unique name, unique number and is located at a single location.



Example of **Company** database...

EMPLOYEE entity

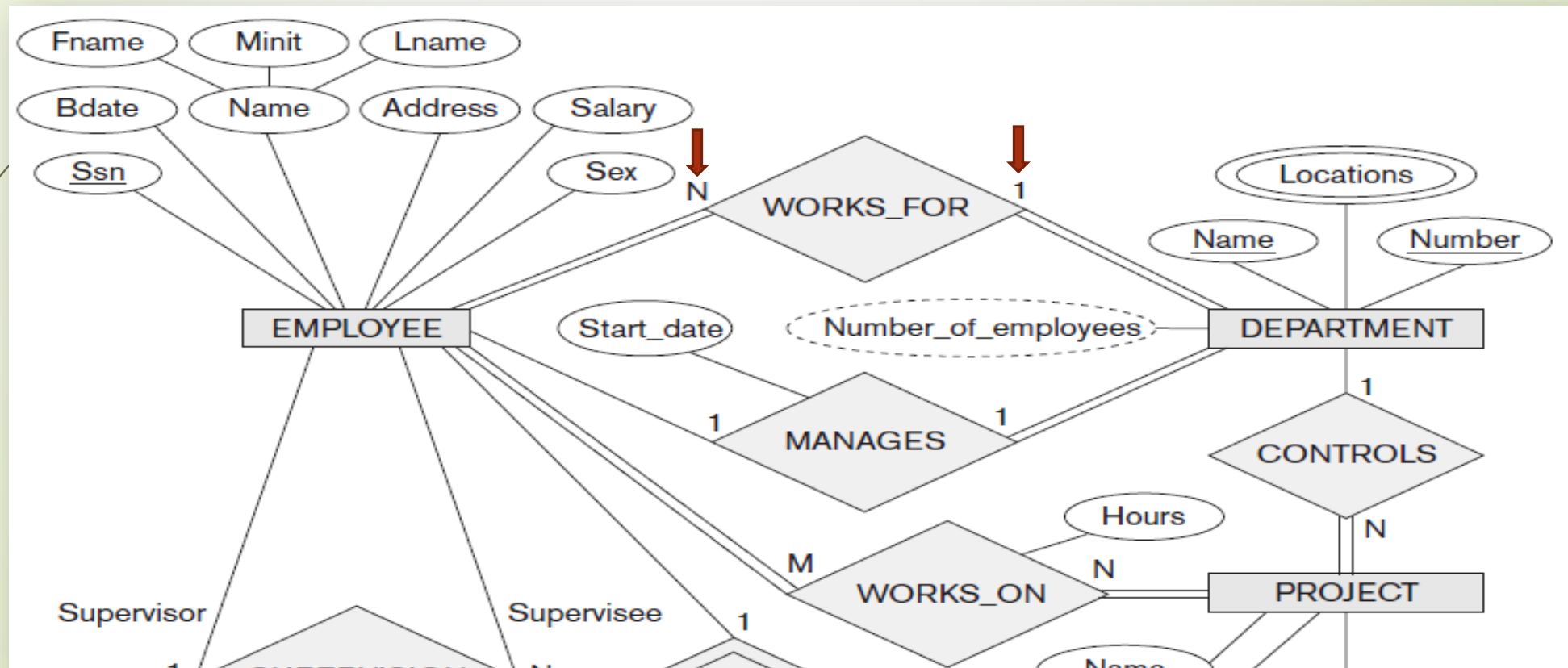
- We store each **EMPLOYEE's** social security number, address, salary, sex, and birthdate



Example of **Company** database...

EMPLOYEE entity...

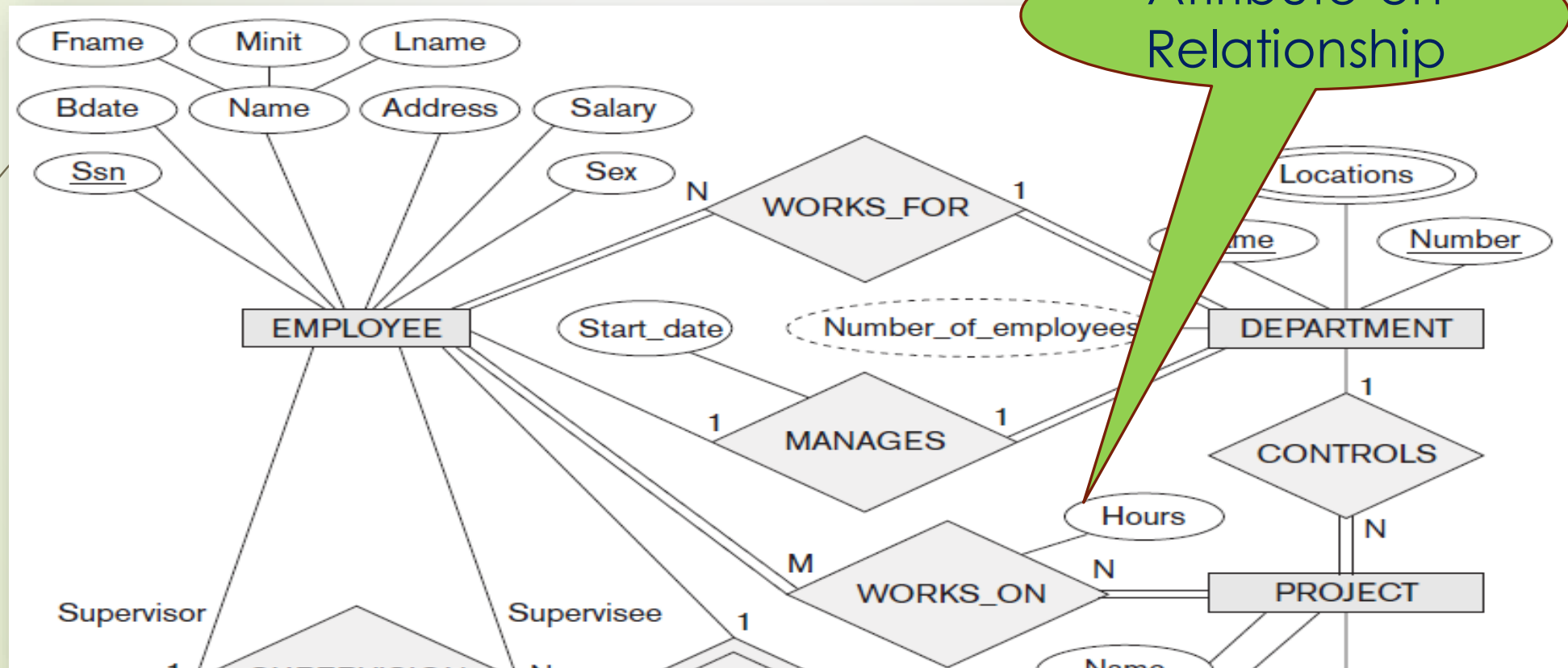
- Each employee *works* for one department, but may *works on* several projects.



Example of **Company** database...

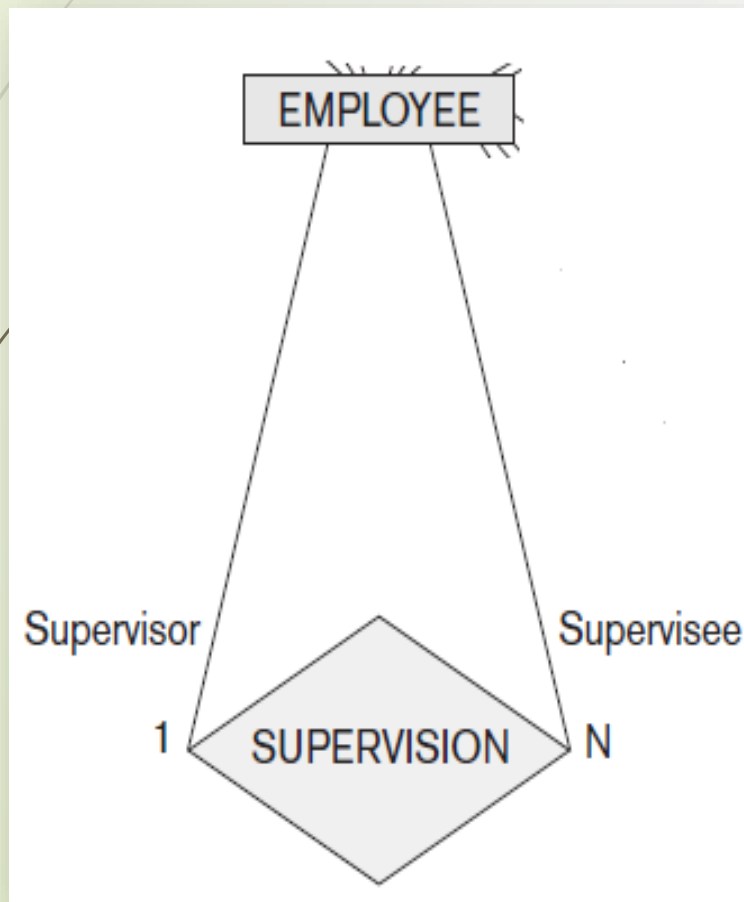
EMPLOYEE entity...

- We keep track **number of hours per week** that an employee currently works on each project.



Example of **Company** database... **EMPLOYEE** entity...

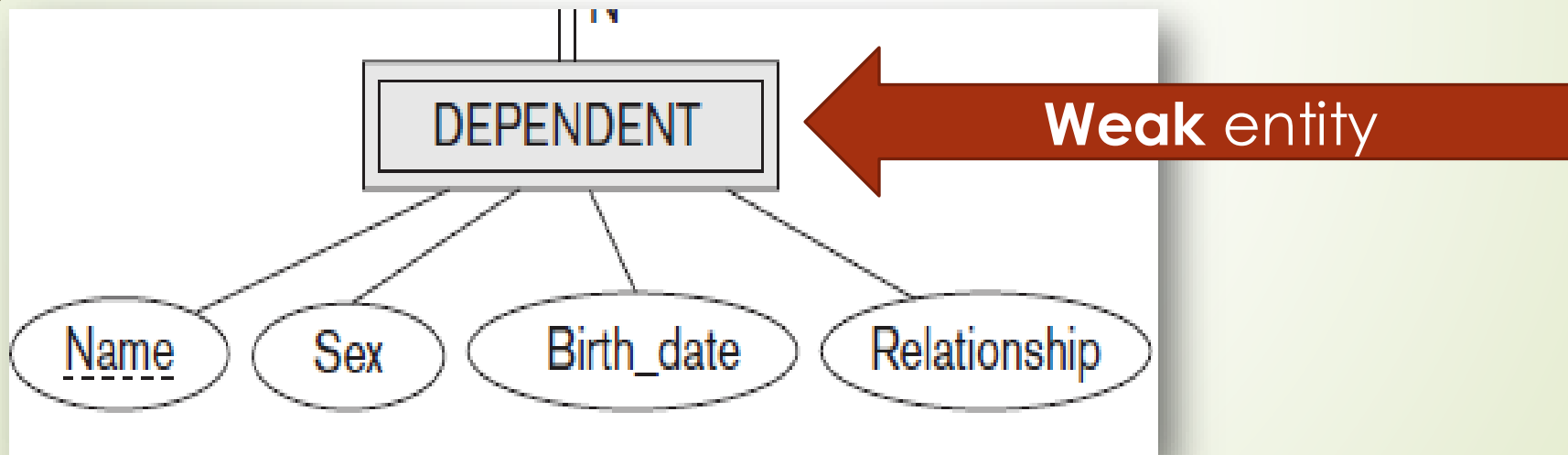
- ➔ We also keep track of the **direct supervisor** of each employee.



Recursive Relationship

Example of **Company** database... **DEPENDENT** entity

- Each employee may have a number of **DEPENDENTS**
 - For each dependent we keep track of their name, sex, birthdate, and relationship to the employee.



Example of **Company** database...

DEPENDENT entity...

► DEPENDENT participation and relation

Note that :

- weak entity usually **total participated** to the owner entity.
- The relation called **identifying relationship**

