

# Chapter

## **Entity Relationship Model Continued**

# RECAP

---

# Main Phases of Database Design

---

1. Requirements Gathering
2. Conceptual Model (high-level)
  - ER Model
3. Logical Model
  - Relational Model
4. Physical Model (not a part of 3530)

# Components of ER Model

---

The “world” is expressed in terms of

- Entity, Entity Type, **Entity Set**
  - **Attribute**
  - Key (Identifier)
  - **Relationship**, Relationship Type, Relationship Set
    - Structural Constraints on relationships
- ER Model is visualized as an ER diagram.

# Example : University Staff Database

---

1. Each department of the University has a unique name, a unique number, and a manager (who is also an employee). For each manager, the start date when he / she was hired as a manager is stored. A department may have several locations.
2. A department runs a number of activities to support its staff and its community, each of which has a name, a unique activity number, and a single location where it is hosted.
3. Each employee's name, social insurance number, address, salary, gender and birth date is stored. An employee works for a department but may volunteer for several activities. We keep track of the number of hours per week that an employee volunteers on each activity. Each employee has a supervisor.
4. An employee may have 0 or more dependents. We keep each dependent's first name, gender, birth date, and relationship to the employee (for insurance purposes).

# Preliminary design of entity types for the University Database

---

Note that {} is used to indicate a multivalued attribute

- DEPARTMENT
  - **dname, dnumber, {locations}, manager, startDate**
- ACTIVITY
  - **aname, anumber, location, department**
- EMPLOYEE
  - **name(FN, LN), SIN, gender, address, salary, dob, department, supervisor, {volunteer(activity, hours)}**
- DEPENDENT
  - **fname, gender, dob, relationship**

# Preliminary design of entity types for the University Staff Database



# Friday Oct 15th

---



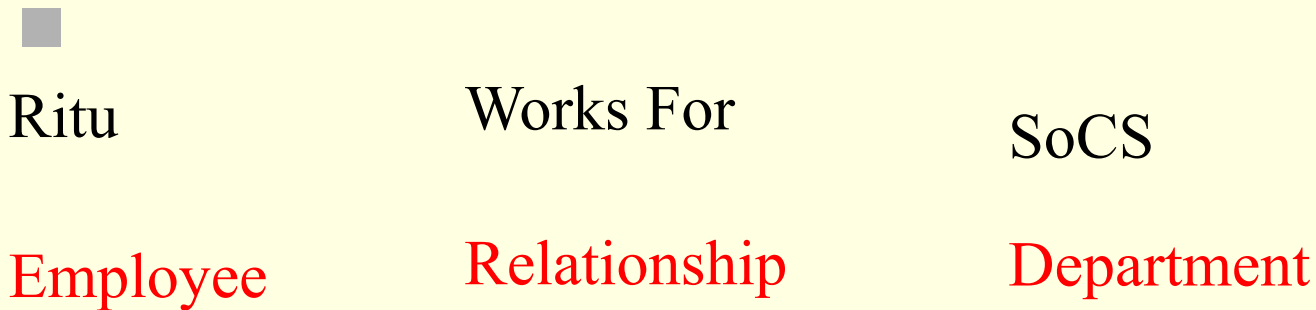
# Worksheet 5 Question 1

---

# Relationship

---

- A relationship is an association among entities.
- Example:



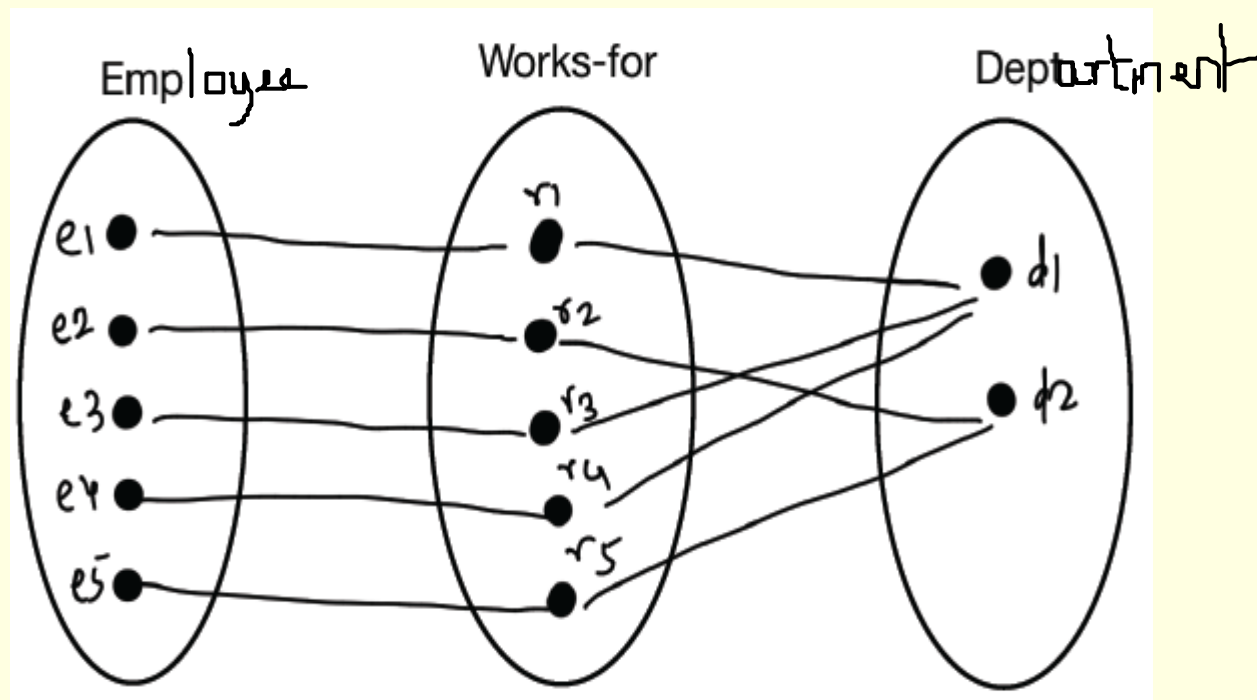
# Example : University Staff Database

---

1. Each **department** of the University has a unique name, a unique number, and a manager (who is also an employee). For each manager, the start date when he / she was hired as a manager is stored. A department may have several locations.
2. A department runs a number of activities to support its staff and its community, each of which has a name, a unique activity number, and a single location where it is hosted.
3. Each employee's name, social insurance number, address, salary, gender and birth date is stored. **An employee works for a department** but may volunteer for several activities. We keep track of the number of hours per week that an employee volunteers on each activity. Each employee has a supervisor.
4. An employee may have 0 or more dependents. We keep each dependent's first name, gender, birth date, and relationship to the employee (for insurance purposes).

# Example – relationship instances

- An employee works for a department



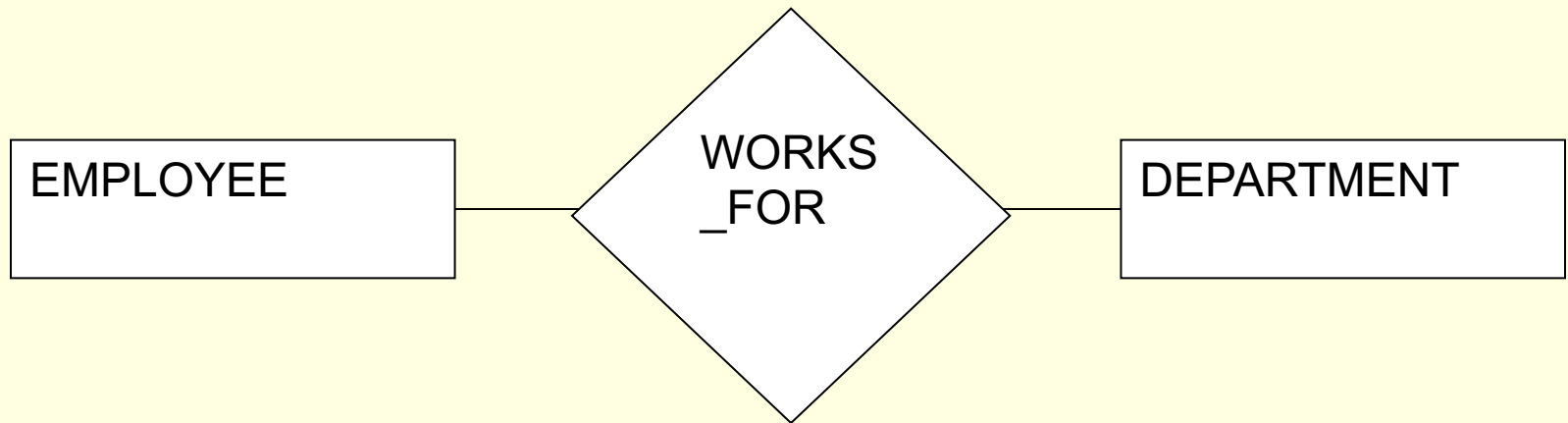
- – note that these instances are only for understanding – how it is represented in the ER model is defined and shown in the next few slides

# Relationship type in ER diagram WORKS\_FOR

---

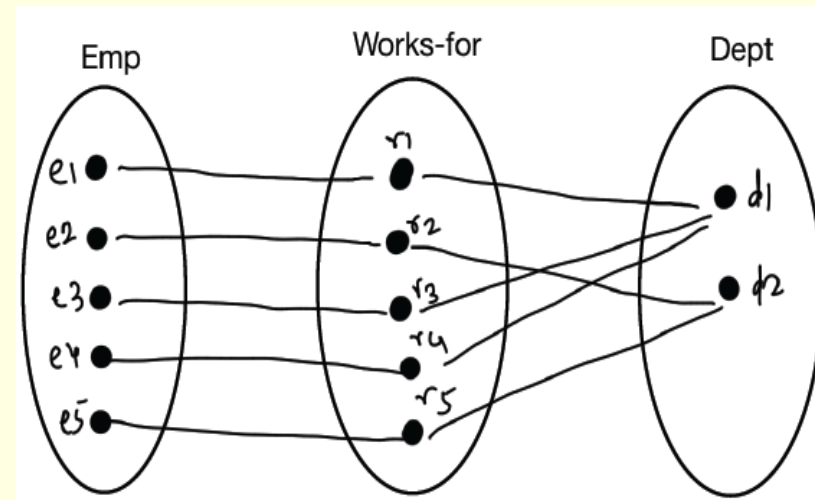
Relationship type: WORKS\_FOR

■ ER :



# Relationship Types

A relationship type  $R$  among  $n$  entity types  $E_1 \dots E_n$  is a set of relationship instances  $r_i$  where each  $r_i$  associates  $n$  individual entities ( $e_1, e_2, \dots, e_n$ ) and each entity  $e_j$  in  $r_i$  is a member of entity type  $E_j$ ,  $1 \leq j \leq n$ .  $E_j$  is called the participating entity type.



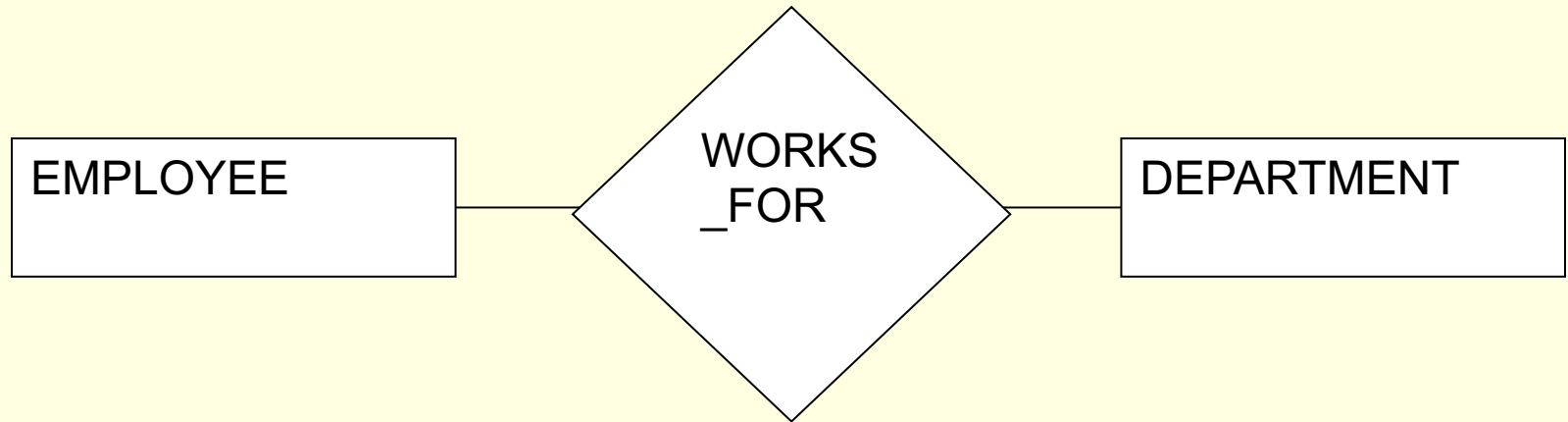
# Example : University Staff Database

---

1. Each department of the University has a unique name, a unique number, and **a manager (who is also an employee)**. For each manager, the start date when he / she was hired as a manager is stored. A department may have several locations.
2. A department runs a number of activities to support its staff and its community, each of which has a name, a unique activity number, and a single location where it is hosted.
3. Each employee's name, social insurance number, address, salary, gender and birth date is stored. **An employee** works for a department but may **volunteer for several activities**. We keep track of the number of hours per week that an employee volunteers on each activity. Each employee has a supervisor.
4. An employee may have 0 or more dependents. We keep each dependent's first name, gender, birth date, and relationship to the employee (for insurance purposes).

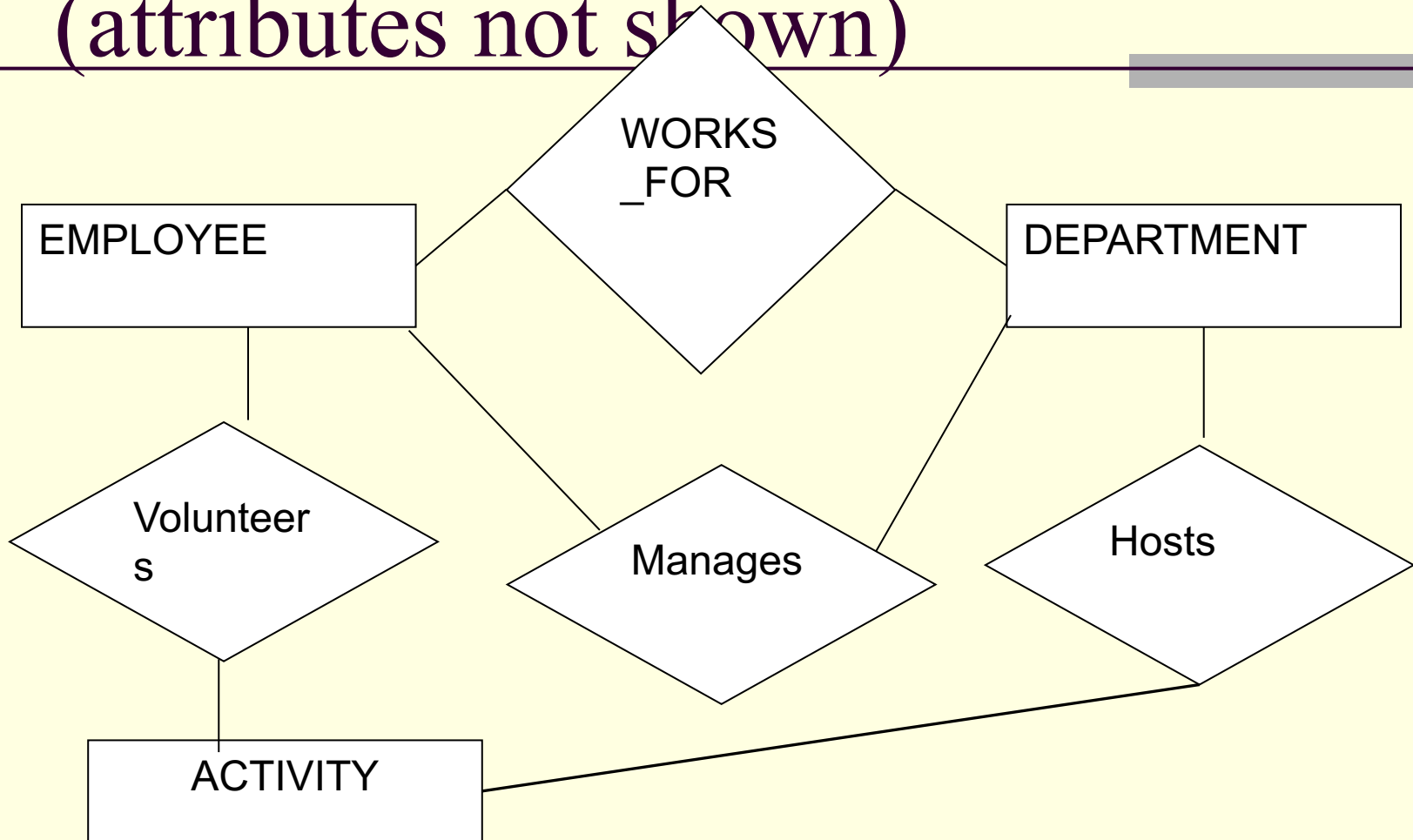
# More relationships

---

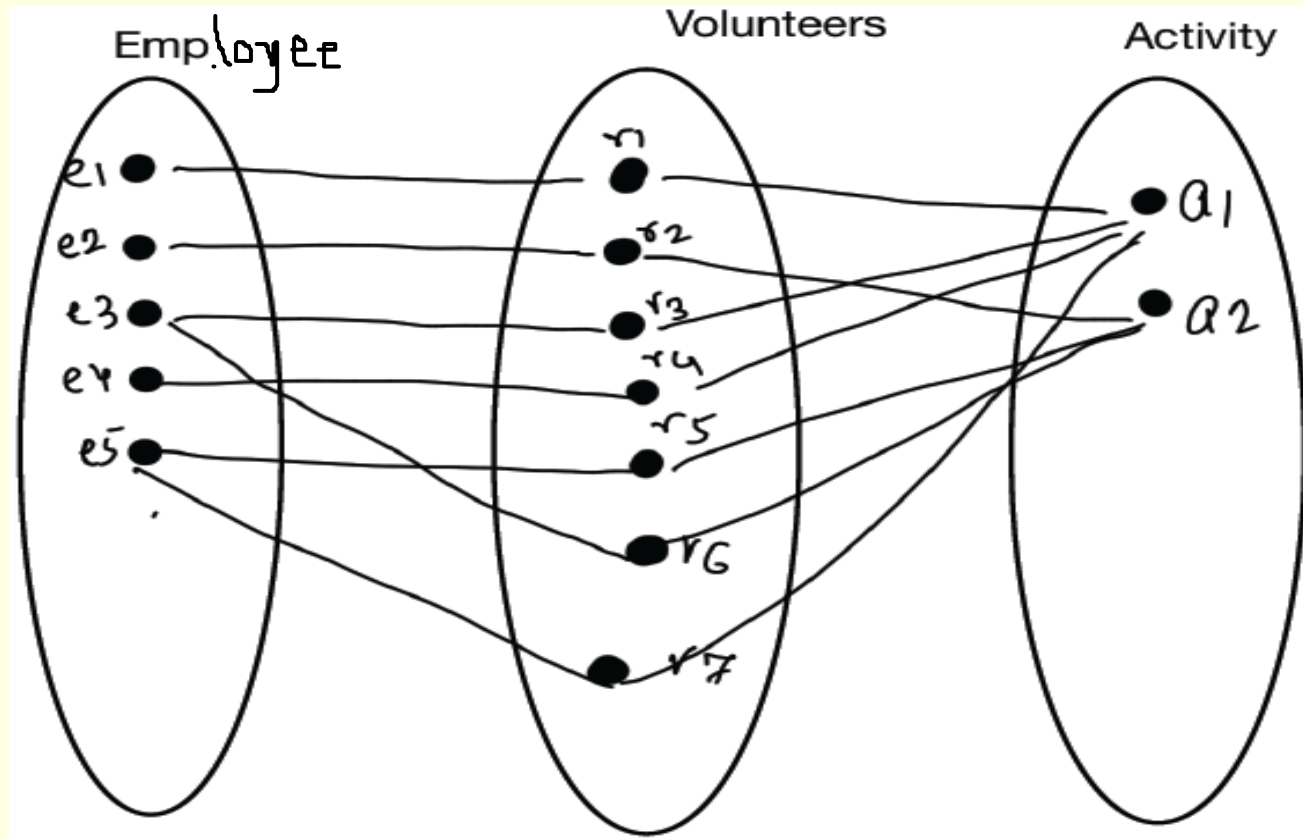




# Some more relationships (attributes not shown)



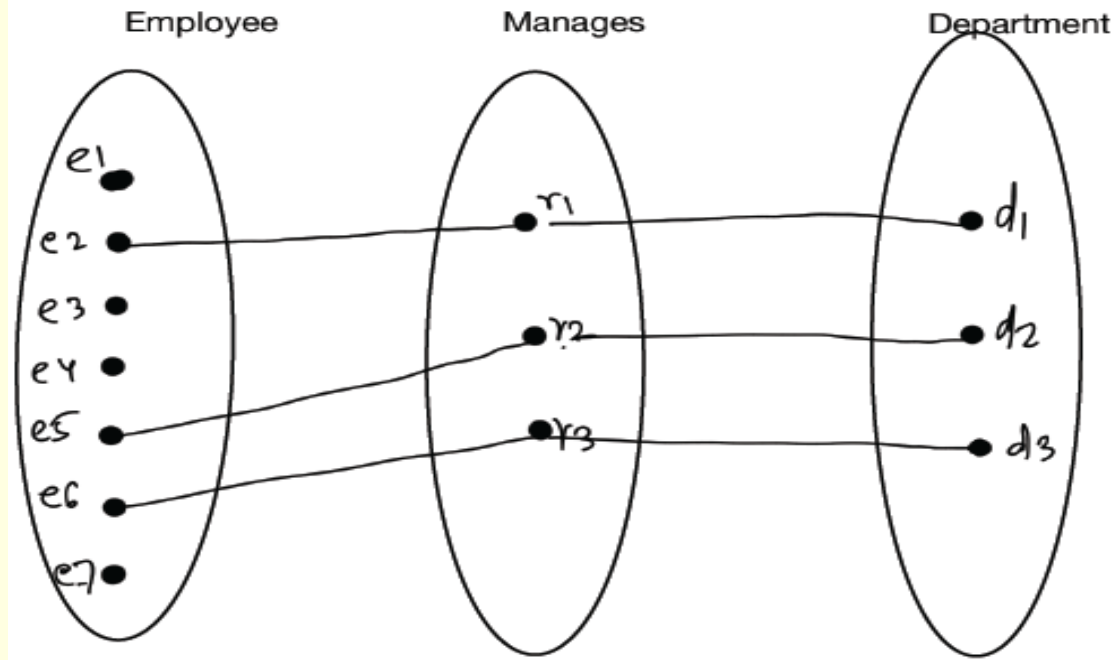
# Relationship type Volunteers



■ ER diagram?

# Relationship type MANAGES

ER Diagram?



# Example : University Staff Database

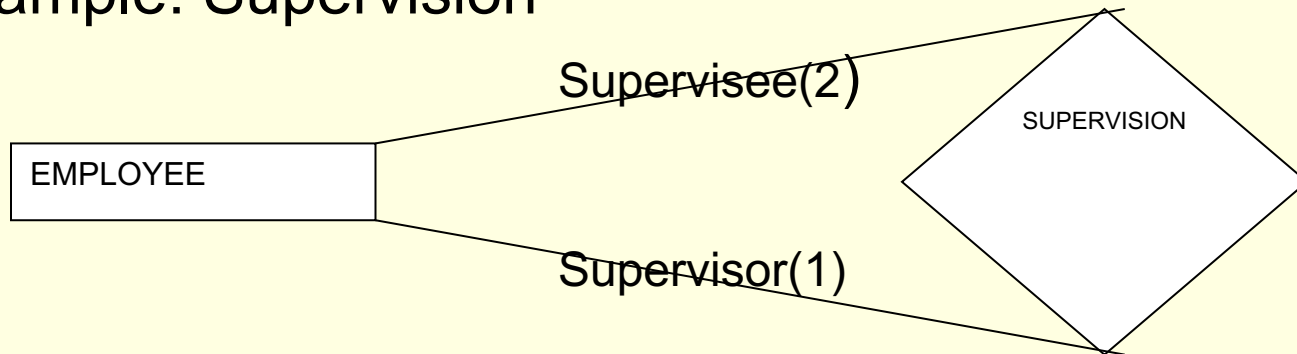
---

1. Each department of the University has a unique name, a unique number, and a manager (who is also an employee). For each manager, the start date when he / she was hired as a manager is stored. A department may have several locations.
2. A department runs a number of activities to support its staff and its community, each of which has a name, a unique activity number, and a single location where it is hosted.
3. Each **employee's** name, social insurance number, address, salary, gender and birth date is stored. An employee works for a department but may volunteer for several activities. We keep track of the number of hours per week that an employee volunteers on each activity. **Each employee has a supervisor.**
4. An employee may have 0 or more dependents. We keep each dependent's first name, gender, birth date, and relationship to the employee (for insurance purposes).

# Recursive Relationships:

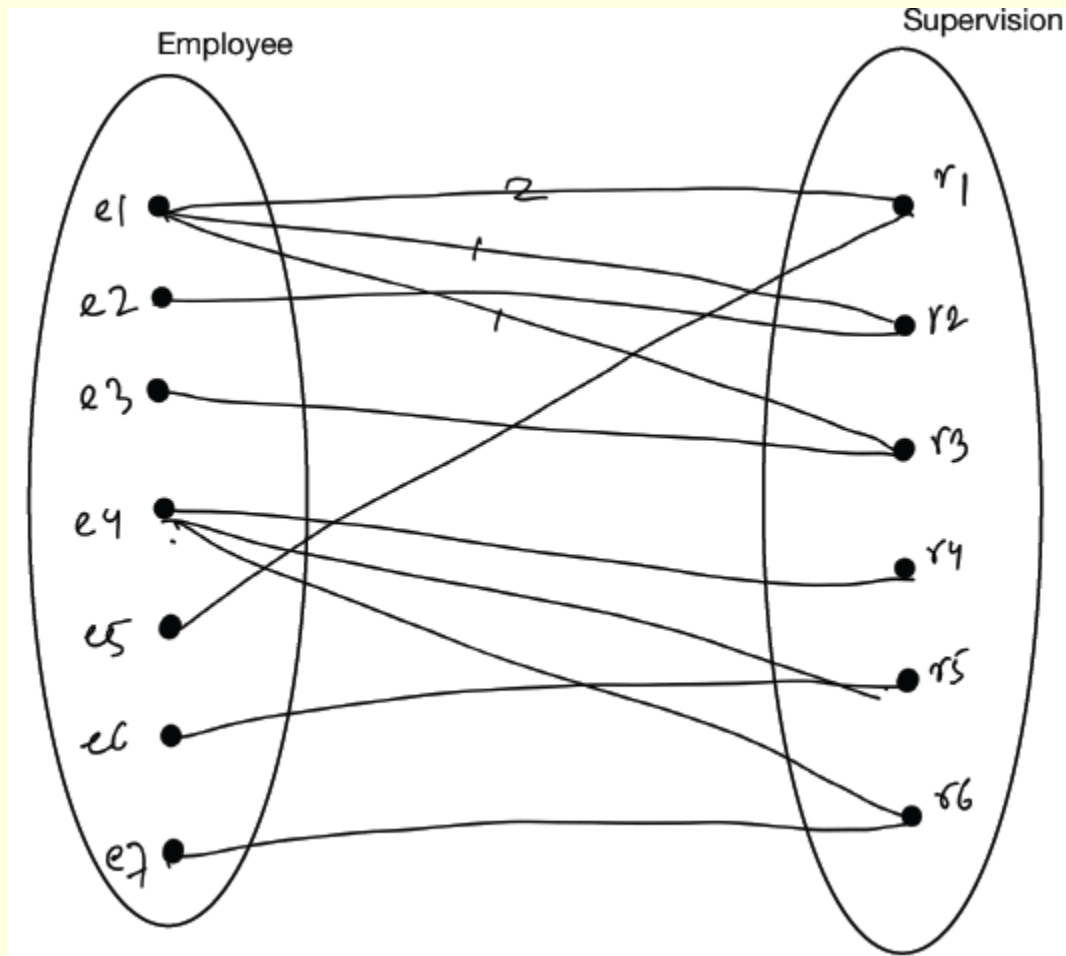
The same entity type participates more than once in a relationship type in different **roles**. Each **role** is given a name.

Example: Supervision



Roles are optional – they clarify semantics of a relationship

# Recursive Relationships



# Examples: recursive relationship

---

- Course and Prerequisites
- People and parents
- Student and Buddy

# Example : University Database:

---

1. Each department of the University has a unique name, a unique number, and a manager (who is also an employee). For each manager, the start date when he / she was hired as a manager is stored. A department may have several locations.
2. A department runs a number of activities to support its students and its community, each of which has a name, a unique activity number, and a single location where it is hosted.
3. Each employee's name, social insurance number, address, salary, gender and birth date is stored. An employee is assigned to one department but may volunteer for several activities. We keep track of the number of hours per week that an employee volunteers on each activity. **Each employee has a supervisor.**
4. **An employee may have 0 or more dependents. We keep each dependent's first name, gender, birth date, and relationship to the employee (for insurance purposes).**



# Weak Entity Type

---

Definition: A weak entity type is one that meets 2 conditions :

1. It is **existence dependent**; that is, it cannot exist without the entity with which it has a relationship.
2. It has a (primary) key that is partially or totally derived from the parent entity in the relationship.

# Weak Entity Type

---

- No key attributes of their own
- Partial key (also called discriminator)
- Identifying Entity Type

ER - 

- Weak entity types always have a total participation constraint with respect to its identifying relationship.

# Example: WET

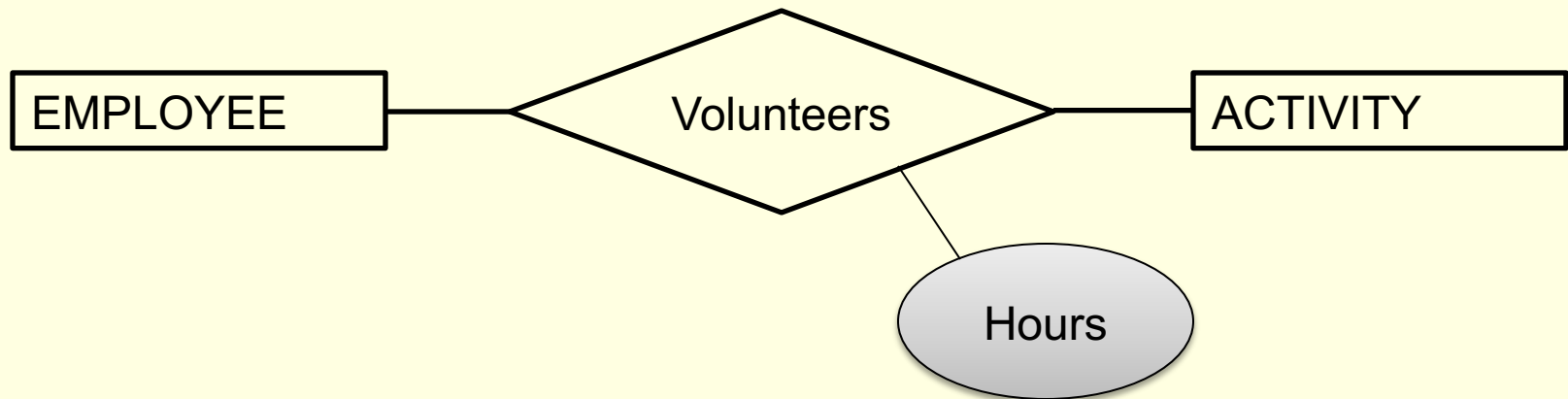
---

- Course and Section (existence of a section has no meaning until it is for a course!)
- Driver and License (license doesn't have a meaning independent of a driver)

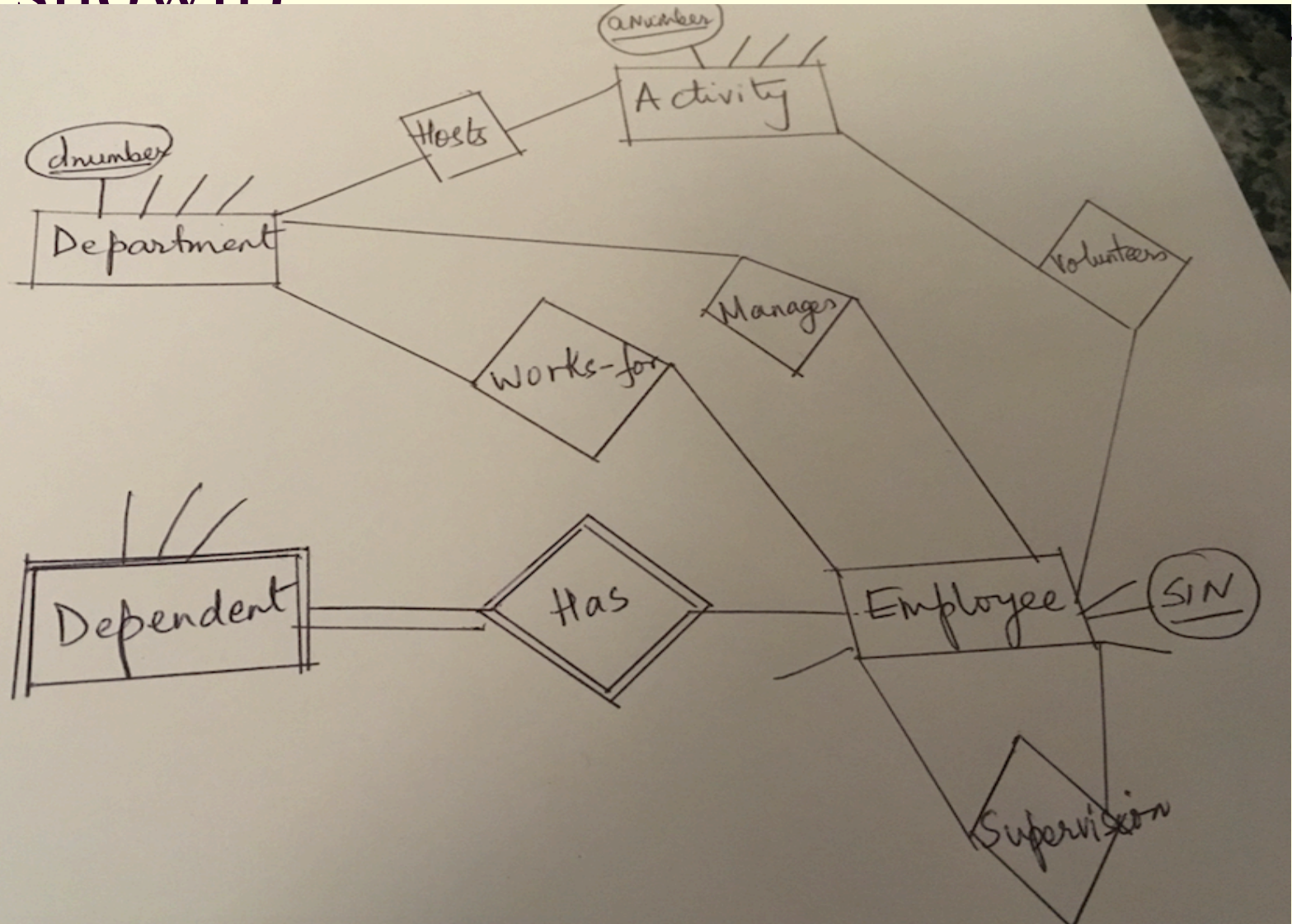
# Attributes of Relationship Types

- A relationship type can have attributes.

Volunteers



# Design so far (only key attributes shown)



# Degree of a relationship

---

Degree of a relationship: Number of relations taking part in a relationship type.

❖ Binary

# Structural Constraints on Relationship Types:

---

- Limit the possible combination of entities
- Represent business rules established by the user
- 2 Structural Constraints :
  - Cardinality Ratio (also known as Relationship Multiplicity)
  - Participation Constraint

# Cardinality Ratio

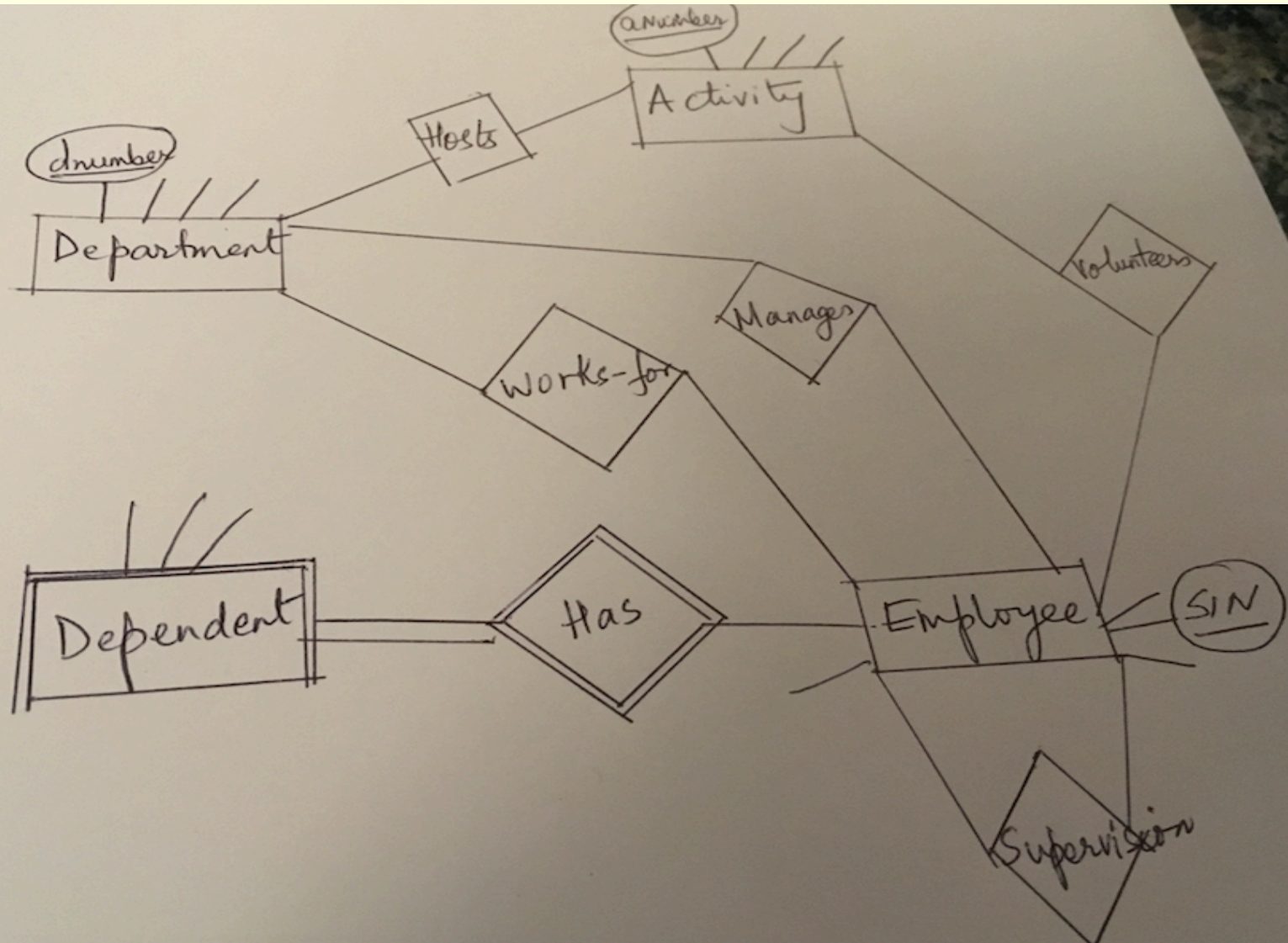
---

Cardinality Ratio : number of relationship instances that an entity can participate in.

- Shown By Placing Appropriate Number On The Link.
  - 1:N, 1:1 , M:N
- Note that the textbook uses an arrow in the direction of 1 side of a relationship (an example is shown in the next slide). I will use Chen's notation mostly



# Design from slide 29 (repeat)



# Worksheet Question 3

---

# Next Week

---

# Participation Constraint

---

Participation Constraint: specifies whether the existence of an entity depends on its being related to another entity via the relationship type.

## **SHOWN BY DOUBLE LINING THE LINK**

- Total (Existence dependency)

ER :        = (double line)

- Partial

# ER model

---

- Entities
  - Strong
  - Weak
- Attributes
  - Simple, composite, single\_valued, multi\_valued, key, relationship attributes
- Relationships
  - Binary
  - Recursive
  - Structural constraints
    - Degree (1:1, 1:N, M:N)
    - Participation (total, partial)