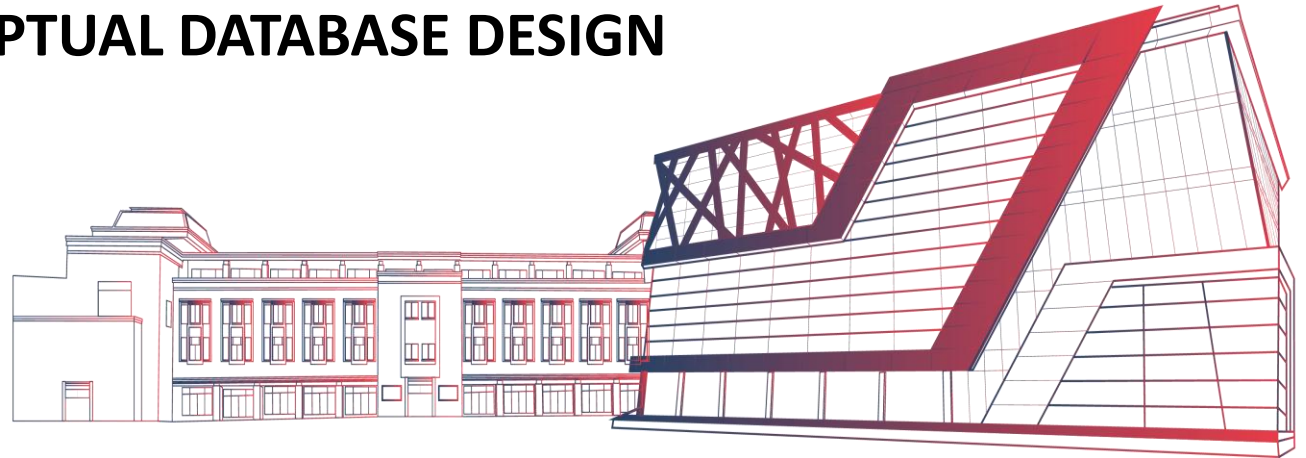


UNIT II

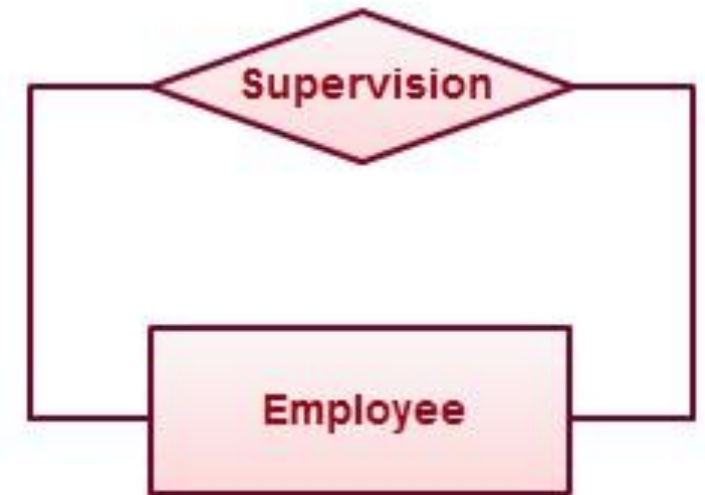
CONCEPTUAL DATABASE DESIGN



Semantics of Relationships

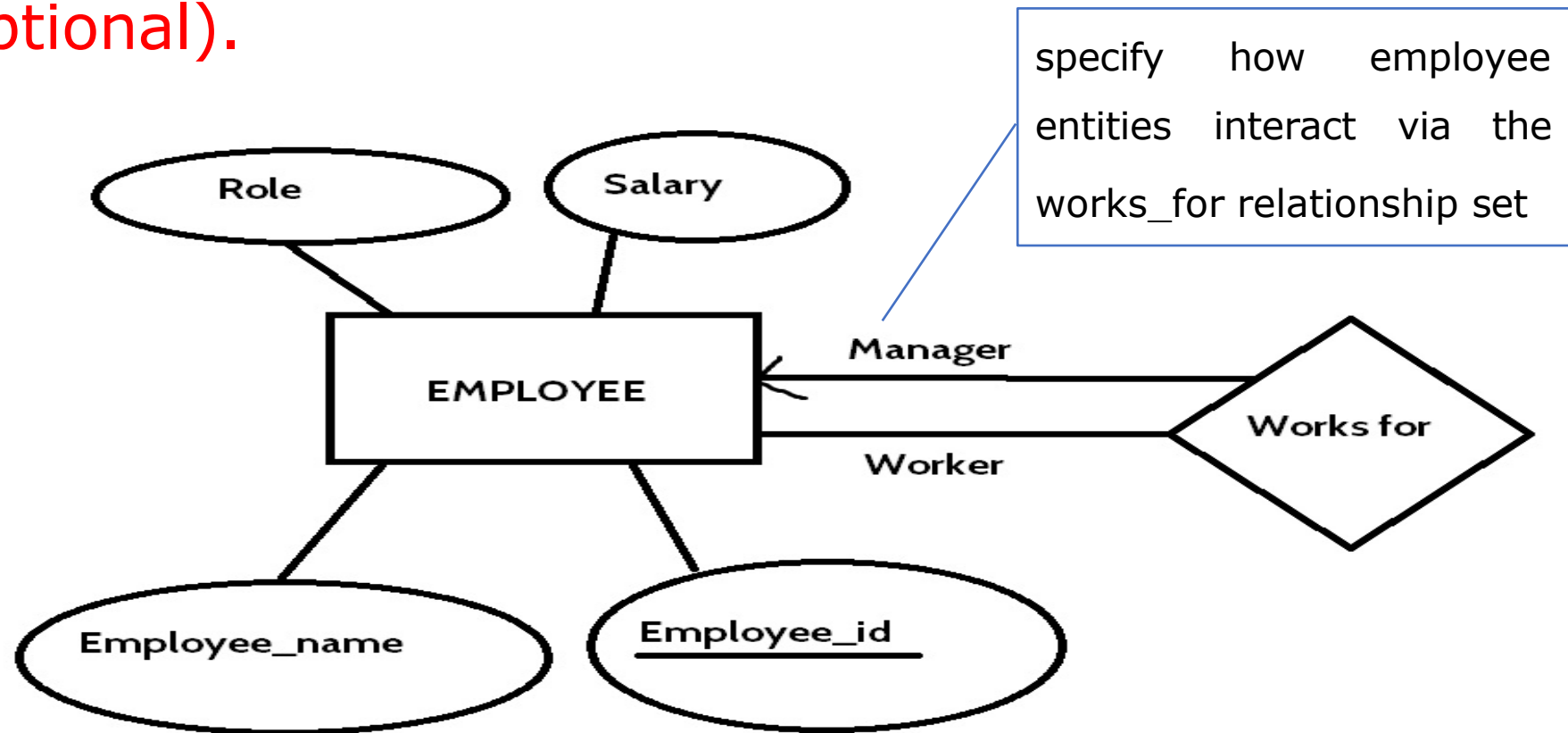
Recursive Relationship

- ▶ An relationship 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



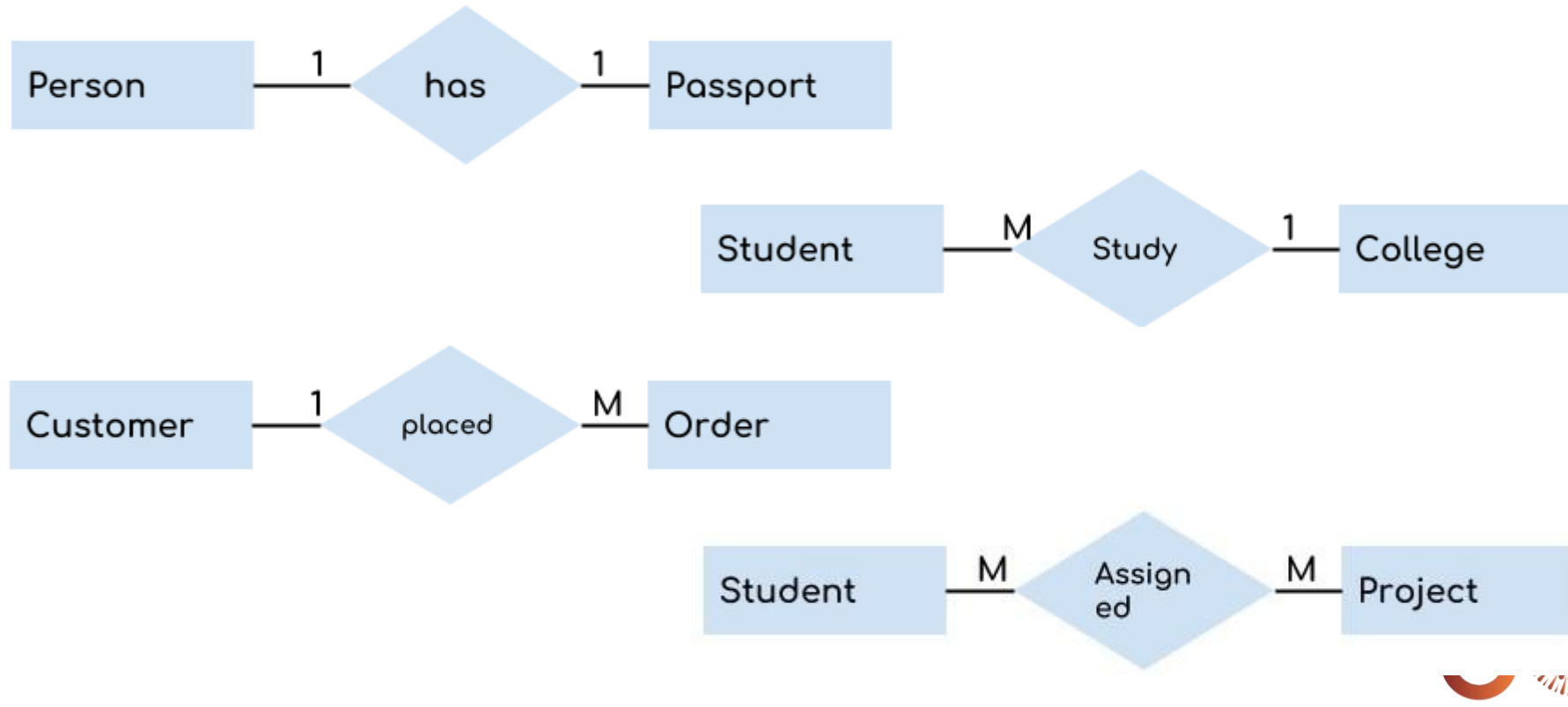
Roles

The function that an entity plays in a relationship is called its **role** (optional).

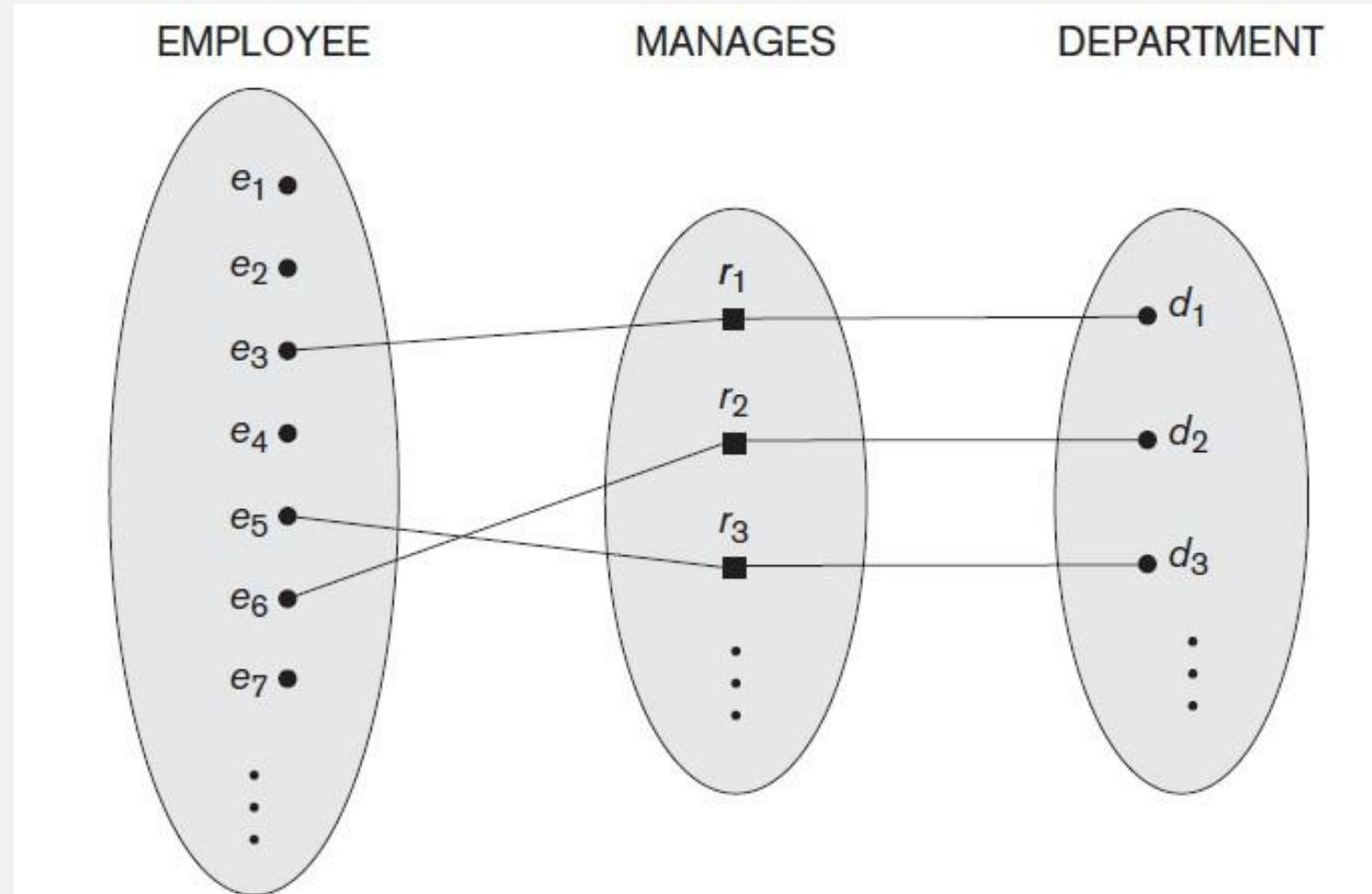


Structural Constraints – Semantics of Relationships

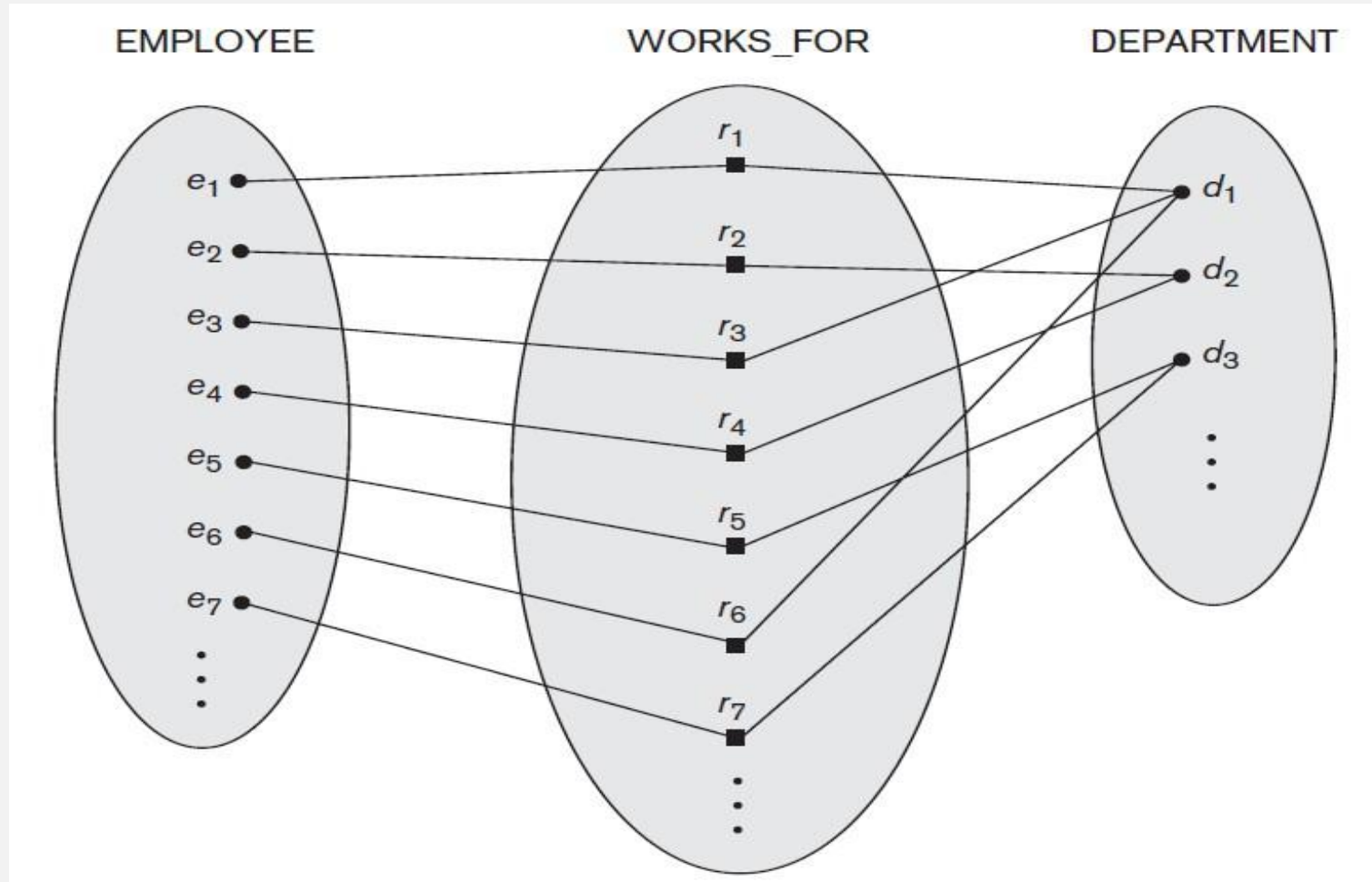
Cardinality Ratio (of a binary relationship): number of instance of an entity from a relation that can be associated with the relation.



1:1 (One-to-One) Relationship - Manages

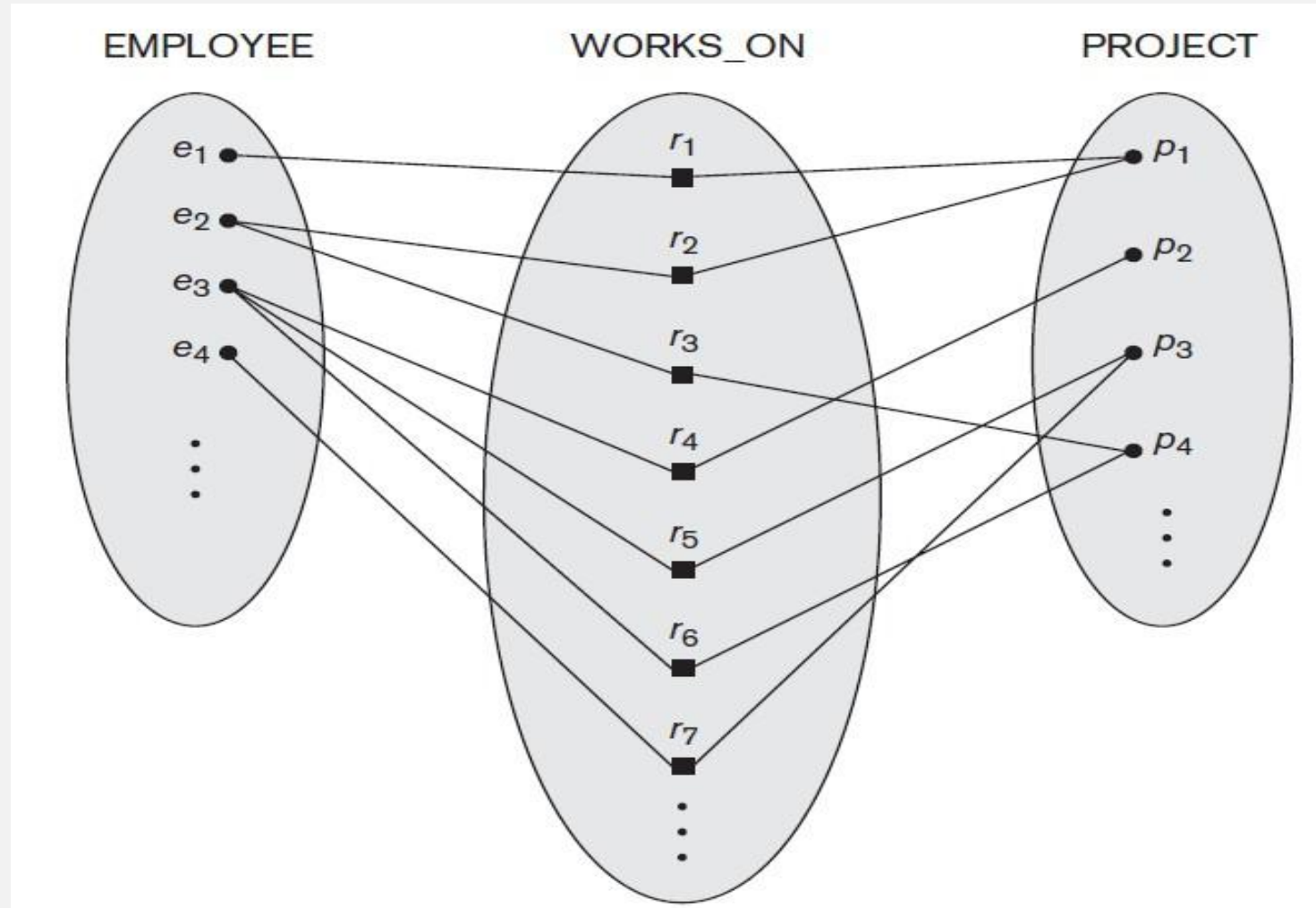


N:1 (Many-to-One) Relationship – Works_for



M:N (Many-to-Many) – Works_on

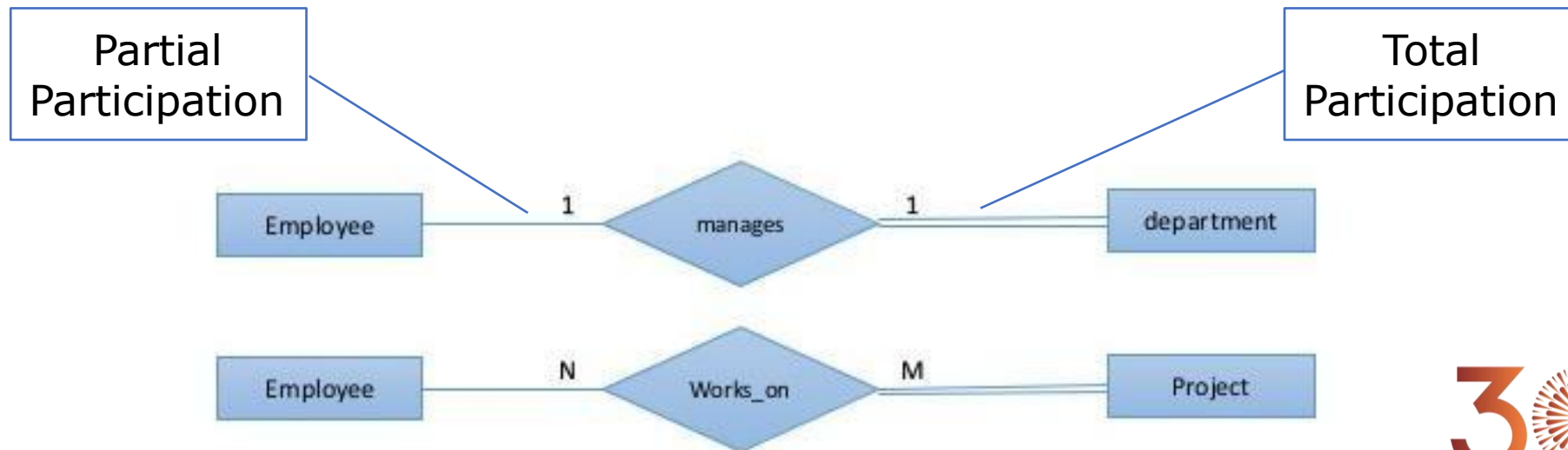
Relationship



Structural Constraints – Semantics of relationships (Cont.)

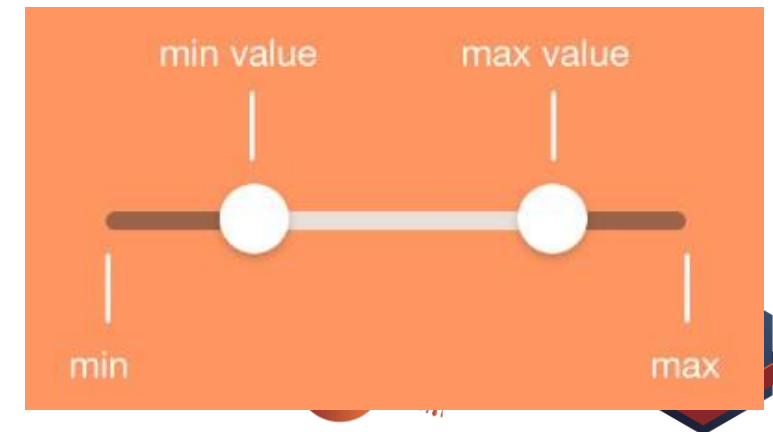
Participation constraint (called *existence dependency*): on each participating entity type

- **Total Participation** – Each entity is involved in the relationship.
- **Partial participation** – Not all entities are involved in the relationship.

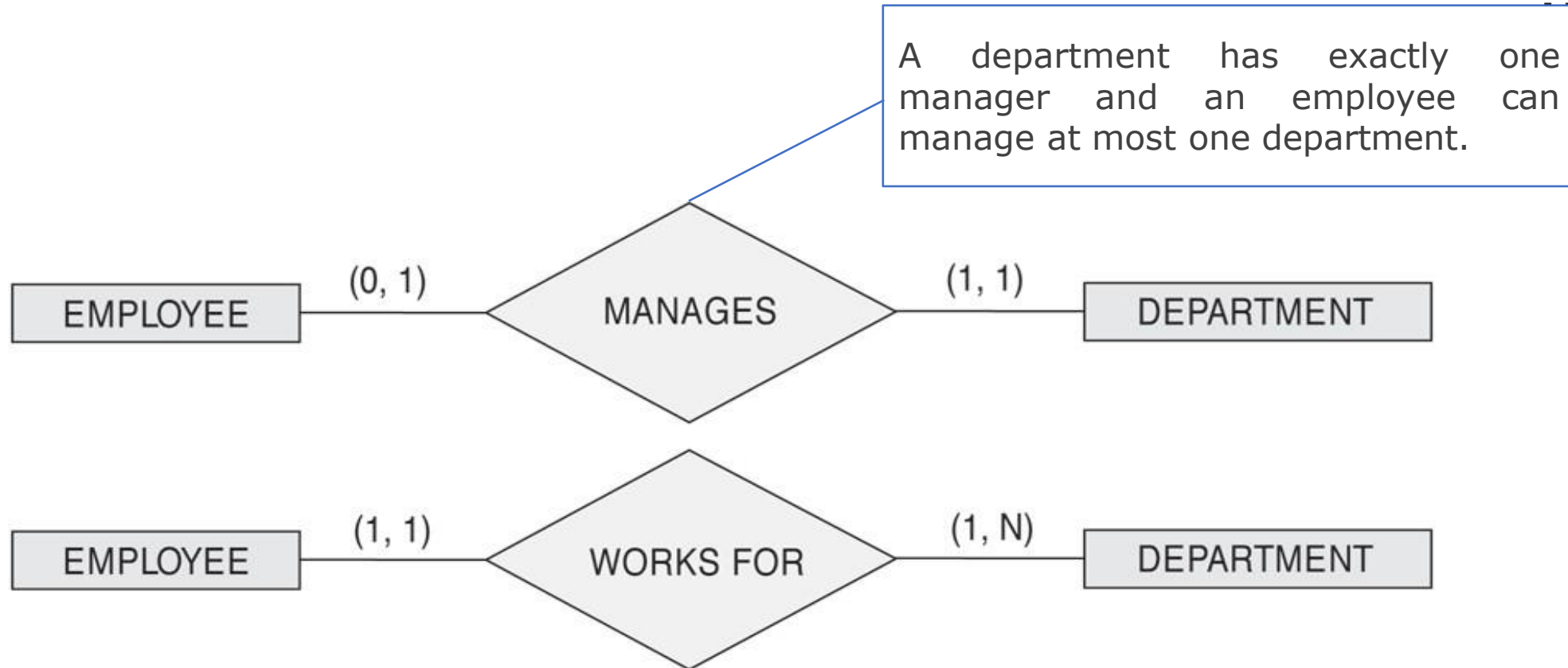


Alternative (min, max) Notation - Relationship Structural Constraints

- ▶ 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): $\text{min}=0, \text{max}=n$ (signifying no limit)
- ▶ Must have $\text{min} \leq \text{max}, \text{min} \geq 0, \text{max} \geq 1$



(Min, Max) Notation - Relationship Constraints

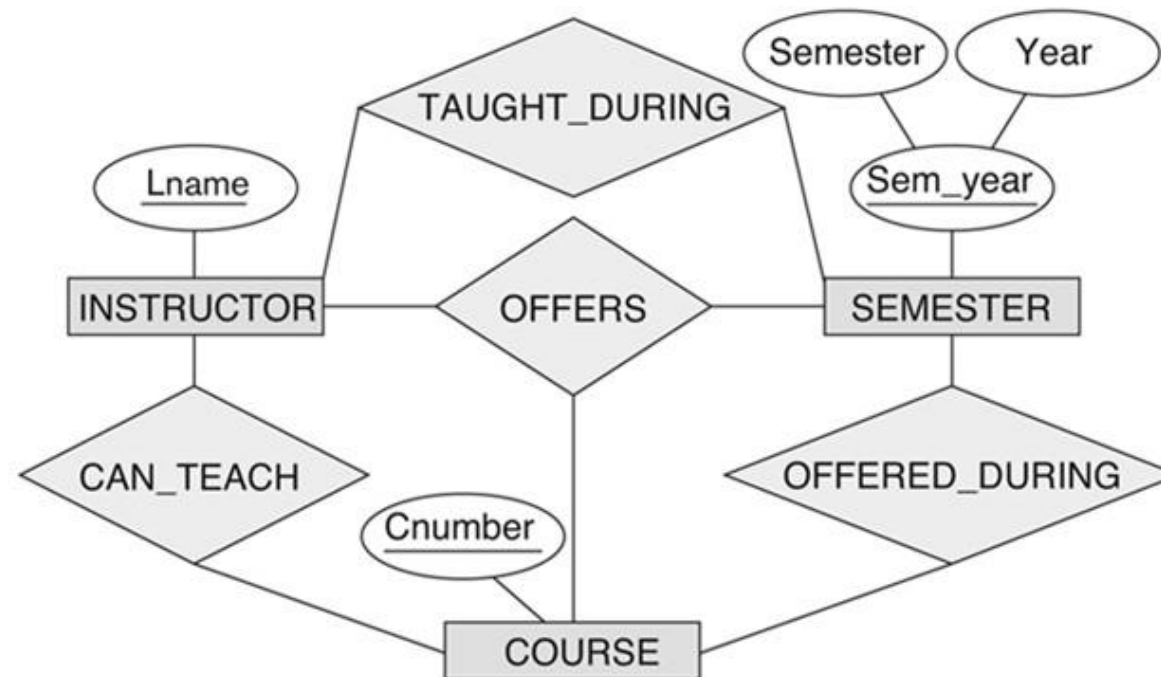


Read the min,max numbers next to the entity type and look away from the entity type

An employee can work for exactly one department but a department can have any number of employees

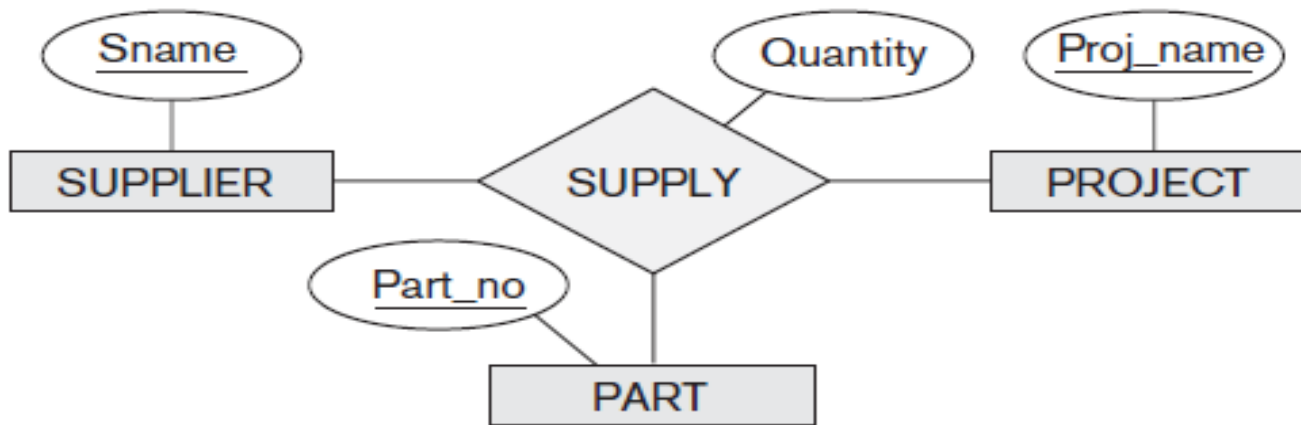
N-ary Relationships ($n > 2$)

- Relationship types of degree 2 are called **binary**
- Relationship types of degree 3 are called **ternary** and of degree n are called **n-ary**



**Ternary
Relationship**

N-ary Relationships (Cont.)

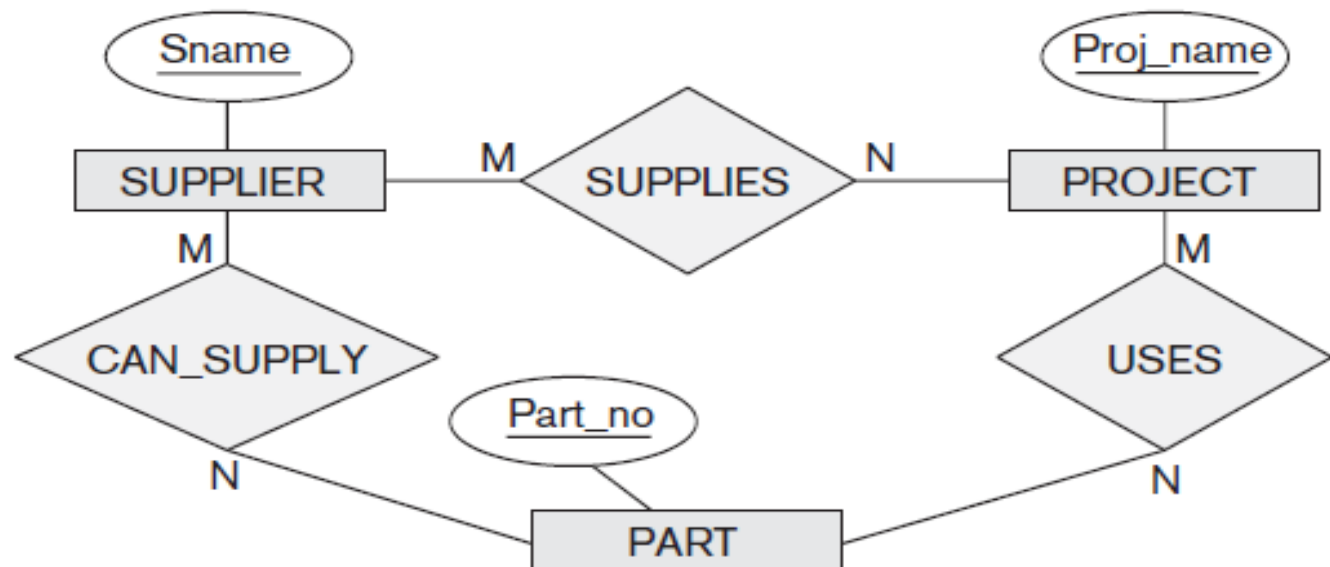


Note:

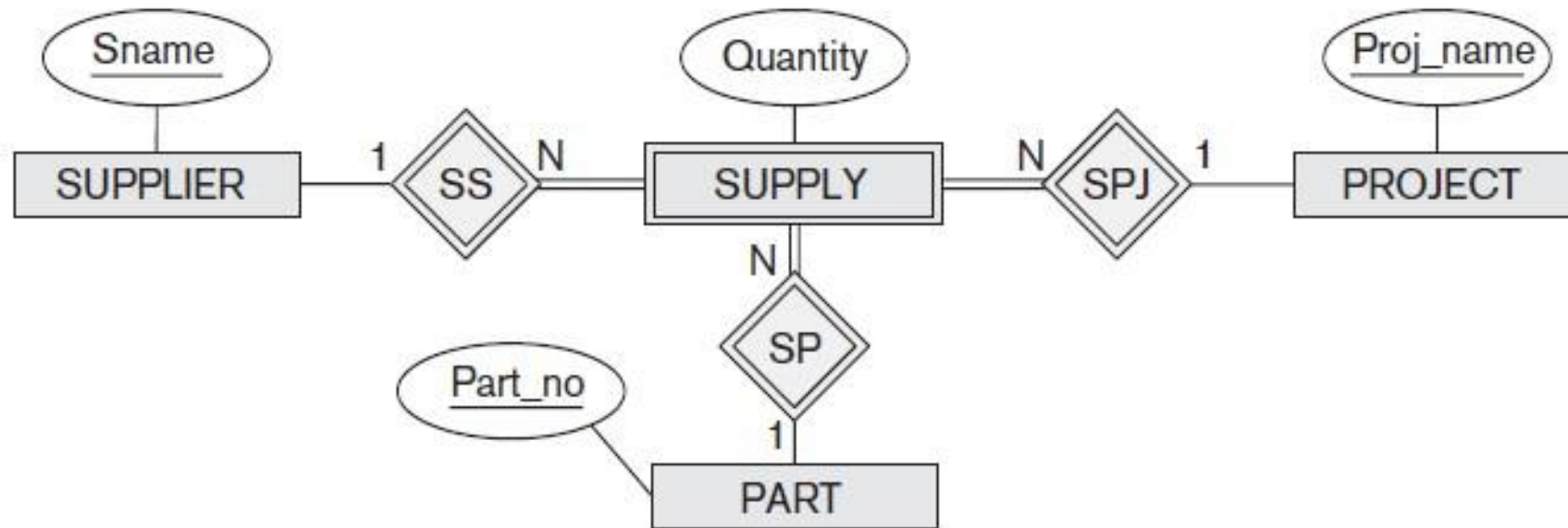
3 binary relationships

≠

1 ternary relationship



N-ary Relationships (Cont.)



**Ternary to Binary relationship – SUPPLY
relationship represented as a weak entity type**

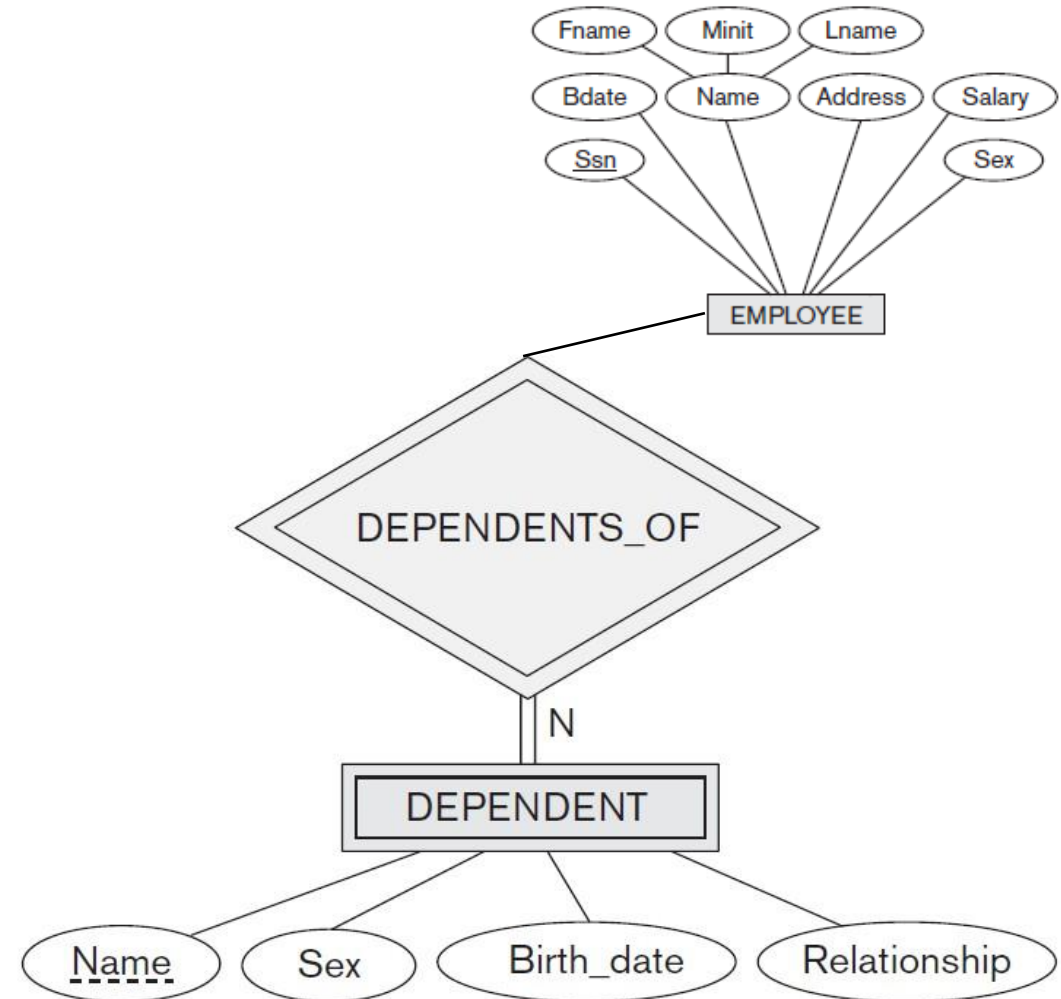
Weak Entity Type

- ▶ **An entity that does not have a key attribute of its own.**
- ▶ The existence of a weak entity set depends on the existence of a **identifying entity set**
 - it must relate to the identifying entity set via a **total, one-to-many relationship set from the identifying to the weak entity set**
 - Identifying relationship depicted using a double diamond
- ▶ 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



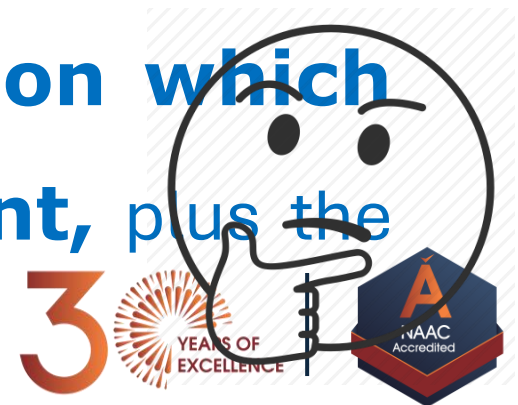
Example: Weak Entity Type

- ▶ 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



How to Distinguish Weak Entity??

- The **discriminator** (*or partial key*) of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set.
- **The primary key of a weak entity set is formed by the primary key of the strong entity set on which the weak entity set is existence dependent, plus the weak entity set's discriminator.**



Thanks!!