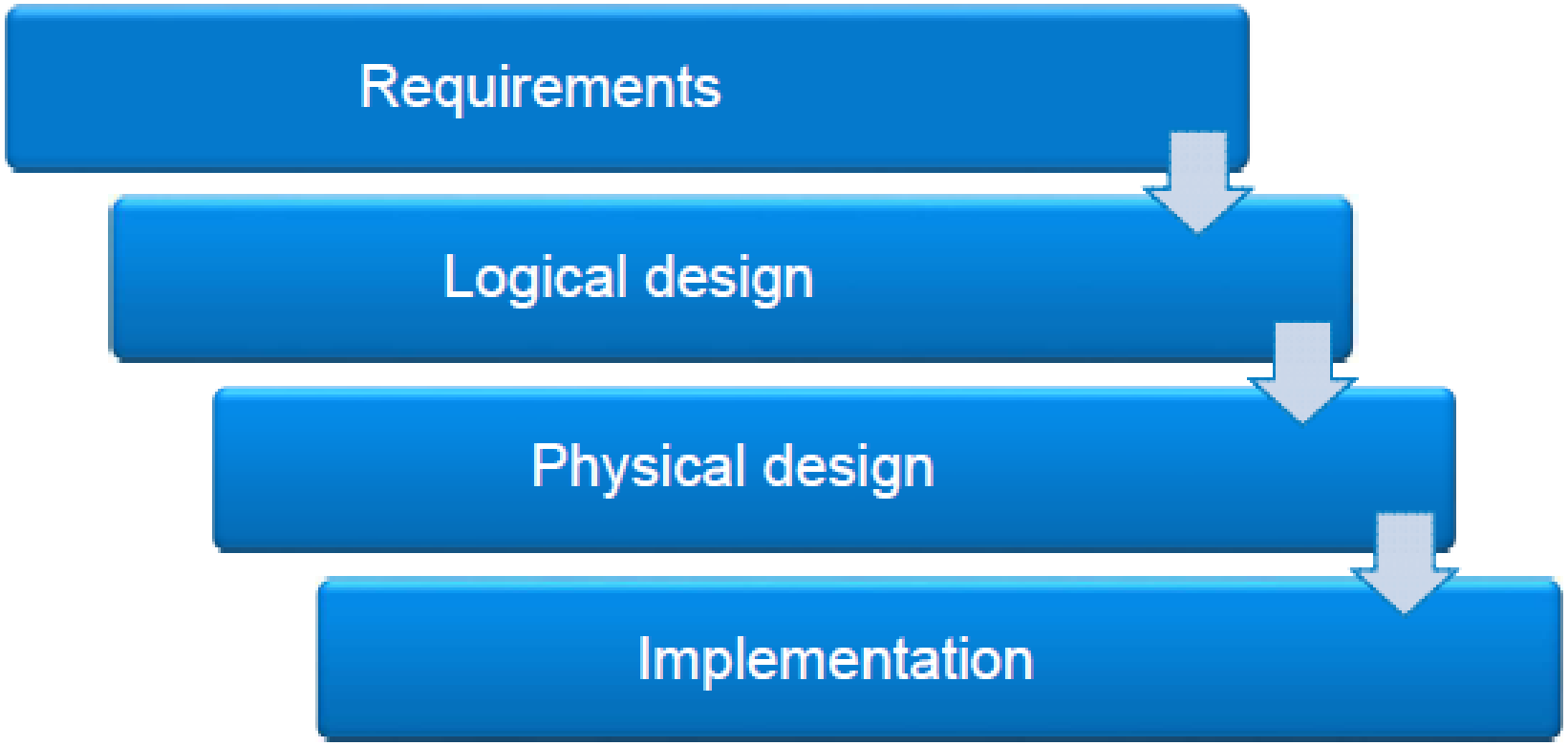
Entity Relationship (ER) Model

Database life cycle



Database life cycle – Data requirements

- This stage involves assessing the informational needs of an organization so that a database can be designed to meet those needs.

Database life cycle – Logical design

Top down approach (Entity – Relationship (ER) model)

- This approach is used when application **requirements are clear**
- Represents the application requirements in a pictorial form
- The real world objects and their corresponding attributes are identified from the requirements – hence it is top down
- This model helps in
  - analysis and design
  - re-validating the requirements

Bottom up approach (Normalization)

- This approach is used when application requirements are not very clear
- First define the required data items and then group the related data items
- Further refinement may be carried out depending on the application need

Symbols used in ER diagram

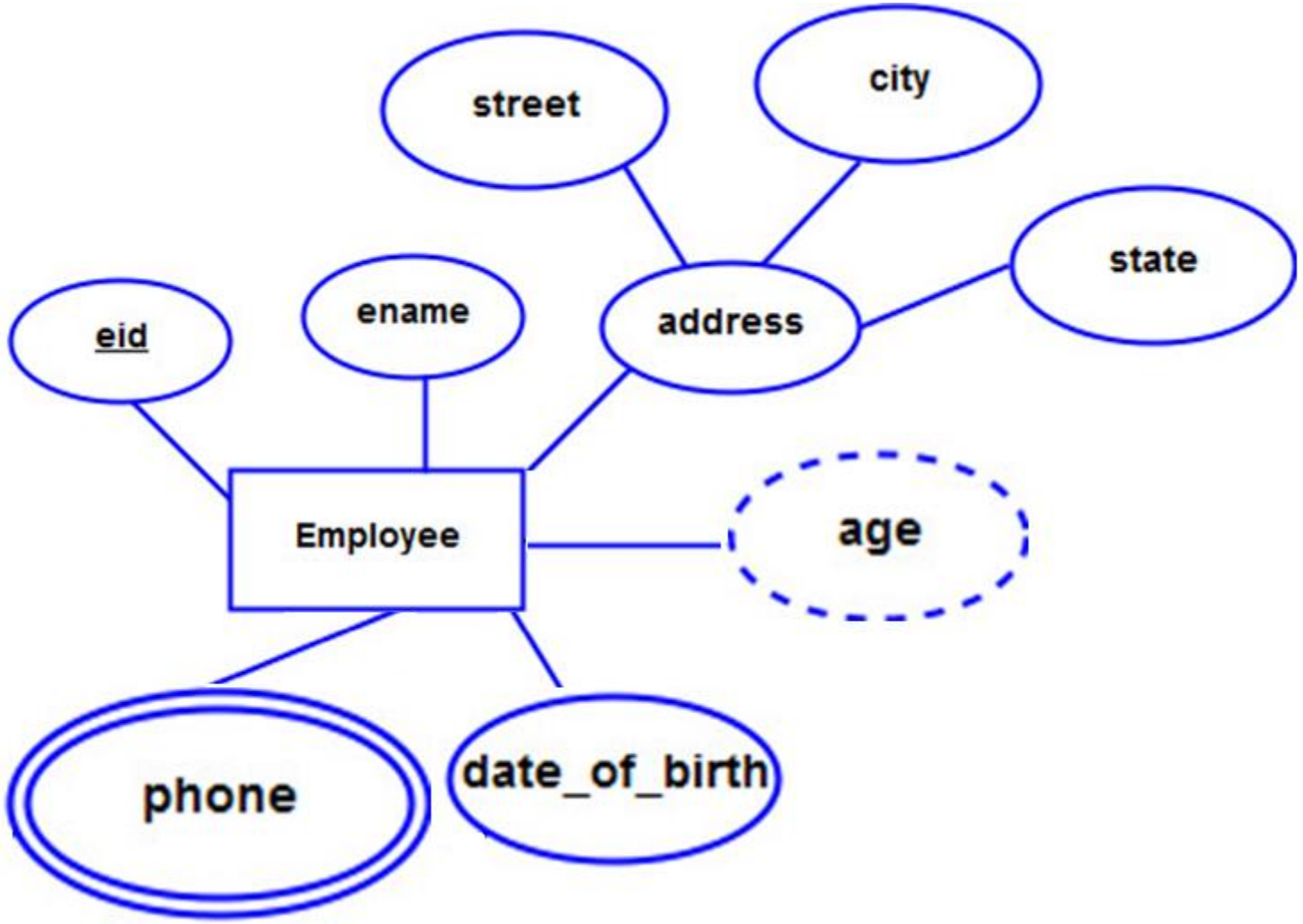
Symbol	Meaning
	ENTITY TYPE
	WEAK ENTITY TYPE
	RELATIONSHIP TYPE
	IDENTIFYING RELATIONSHIP TYPE
	ATTRIBUTE
	KEY ATTRIBUTE
	MULTIVALUED ATTRIBUTE
	COMPOSITE ATTRIBUTE
	DERIVED ATTRIBUTE
E_1 — R ═ E_2	TOTAL PARTICIPATION OF E_2 IN R
E_1 — 1 — R — N — E_2	CARDINALITY RATIO 1:N FOR E_1:E_2 IN R
— R — (min,max) — E	STRUCTURAL CONSTRAINT (min, max) ON PARTICIPATION OF E IN R

Entity and Entity type

- **Entities** - are specific objects or things in the mini-world that are represented in the database.
- Eg: the EMPLOYEE John Smith, the Research DEPARTMENT, the ProductX PROJECT

- Entities with the same basic attributes are grouped or typed into an **entity type**.
- Eg; EMPLOYEE is an entity type

Entity type and attributes



Attributes

- **Attributes** are properties used to describe an entity.
- For example, an EMPLOYEE entity may have a Name, SSN, Address, Sex, BirthDate are 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-2005’

Types of Attributes

- **Single Vs Composite**

- **Simple**: Each entity has a single atomic value for the attribute
- For example: SSN

- **Composite**: The attribute may be composed of several components.
- For example: Name(FirstName, MiddleName, LastName).

Types of Attributes(cont)

- **Single-valued Vs Multi-valued**

- **Single-valued**: An entity must have only one value for that attribute.
- For example: SSN of an EMPLOYEE

- **Multi-valued**: An entity may have multiple values for that attribute.
- For example, PreviousDegrees of a STUDENT.

Types of Attributes(cont)

- **Stored Vs Derived**

- **Stored**: An attribute that is stored in the database.
- For example: SSN of an EMPLOYEE

- **Dervided**: An attribute that is derived from the stored attribute.
- For example, Age of an EMPLOYEE. Age is derived from Date-of-birth

Types of Attributes(cont)

- **Complex attribute**

- **Complex** : An attribute which is both composite and multi-valued
- Eg Assume an Employee has more than one address. Address is composite (street, city, zipcode)

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.

- An entity type may have more than one key.

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

- 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.

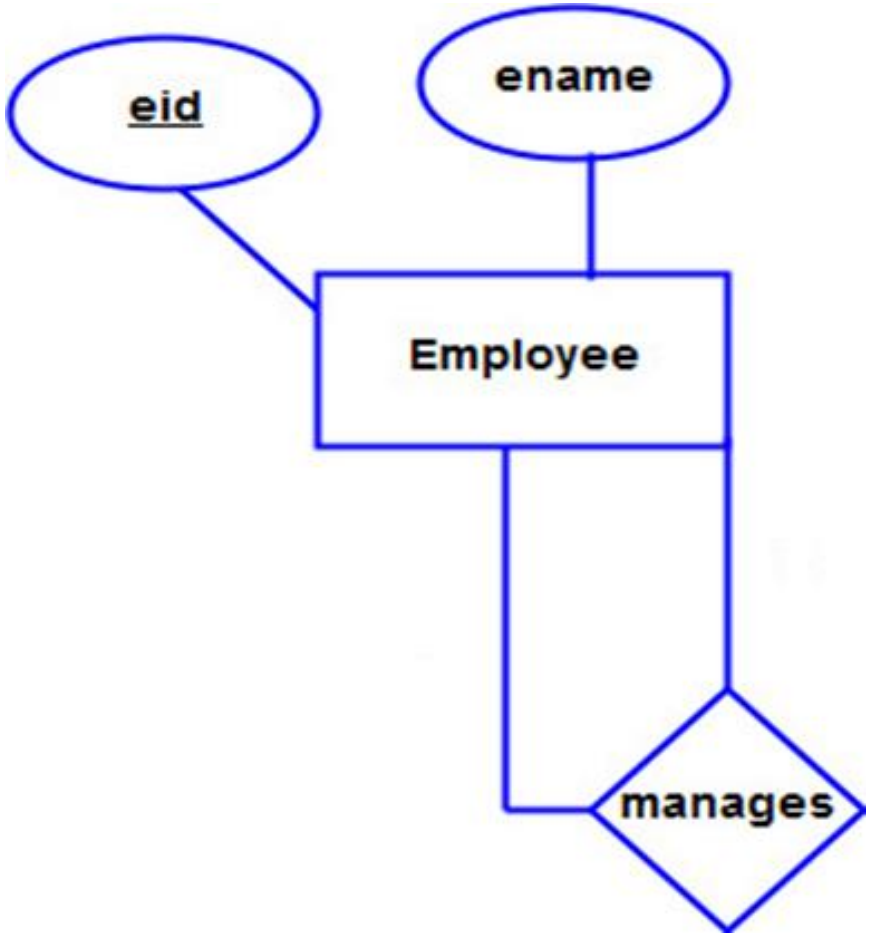
- More than one relationship type can exist with the same participating entity types.

Degree of a relationship type

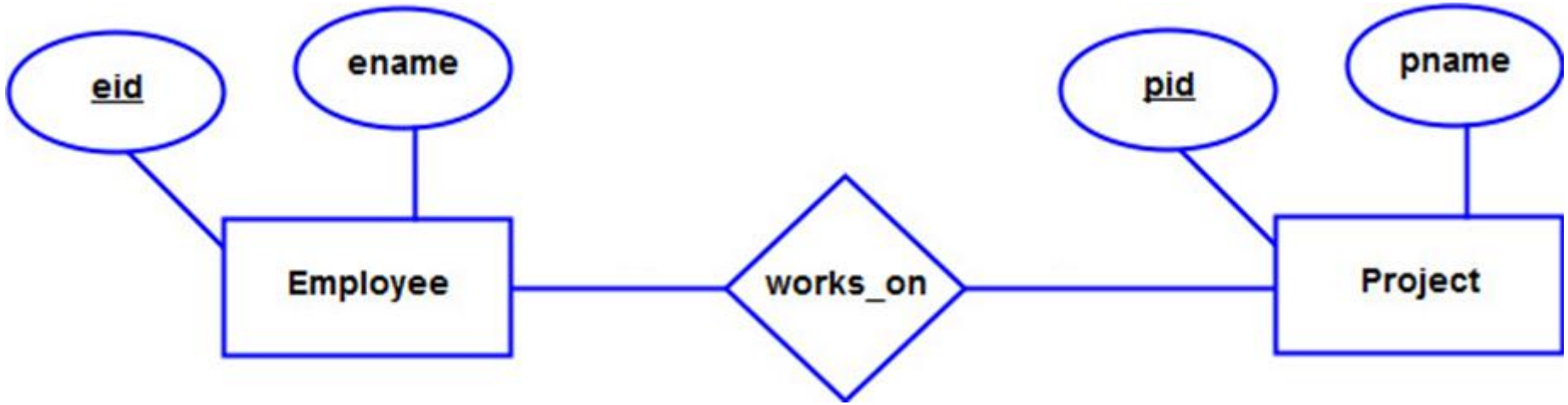
- The number of entity types participating in a relationship type.

- Degree may be 1 (unary relationship type)
- Degree may be 2 (binary relationship type)
- Degree may be 3 (ternary relationship type)
- Degree may be more than 3 (n-ary relationship type)

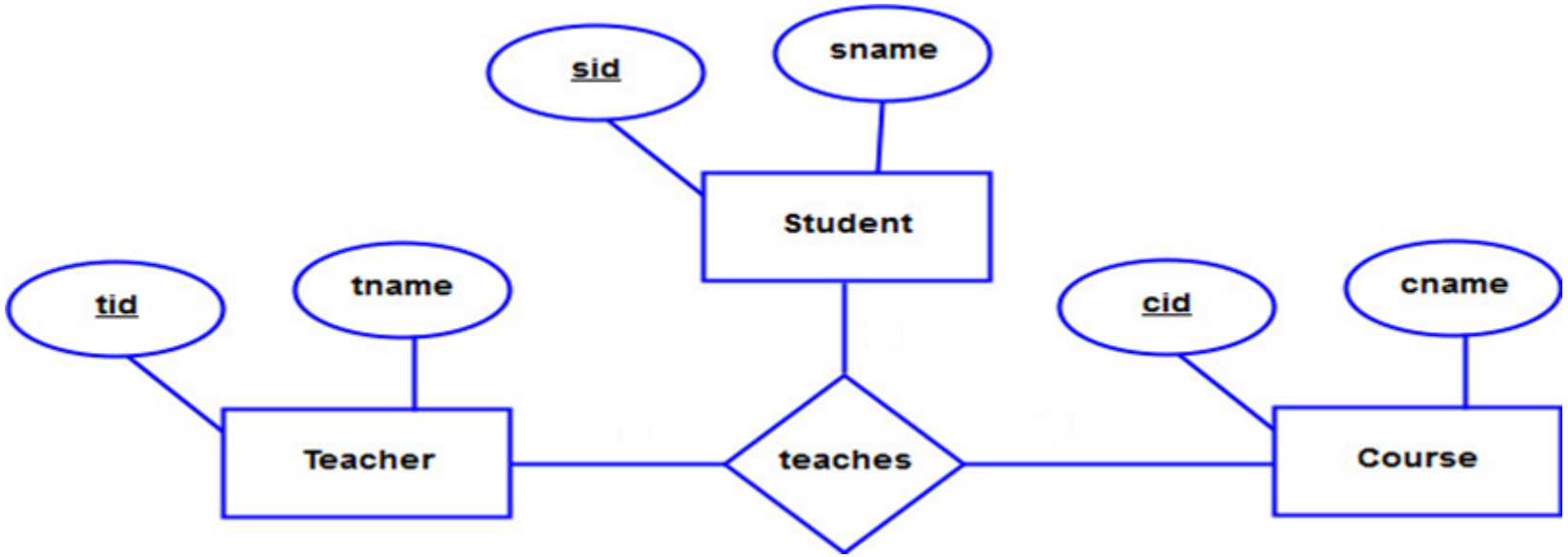
unary relationship type



Binary relationship type



Ternary relationship type

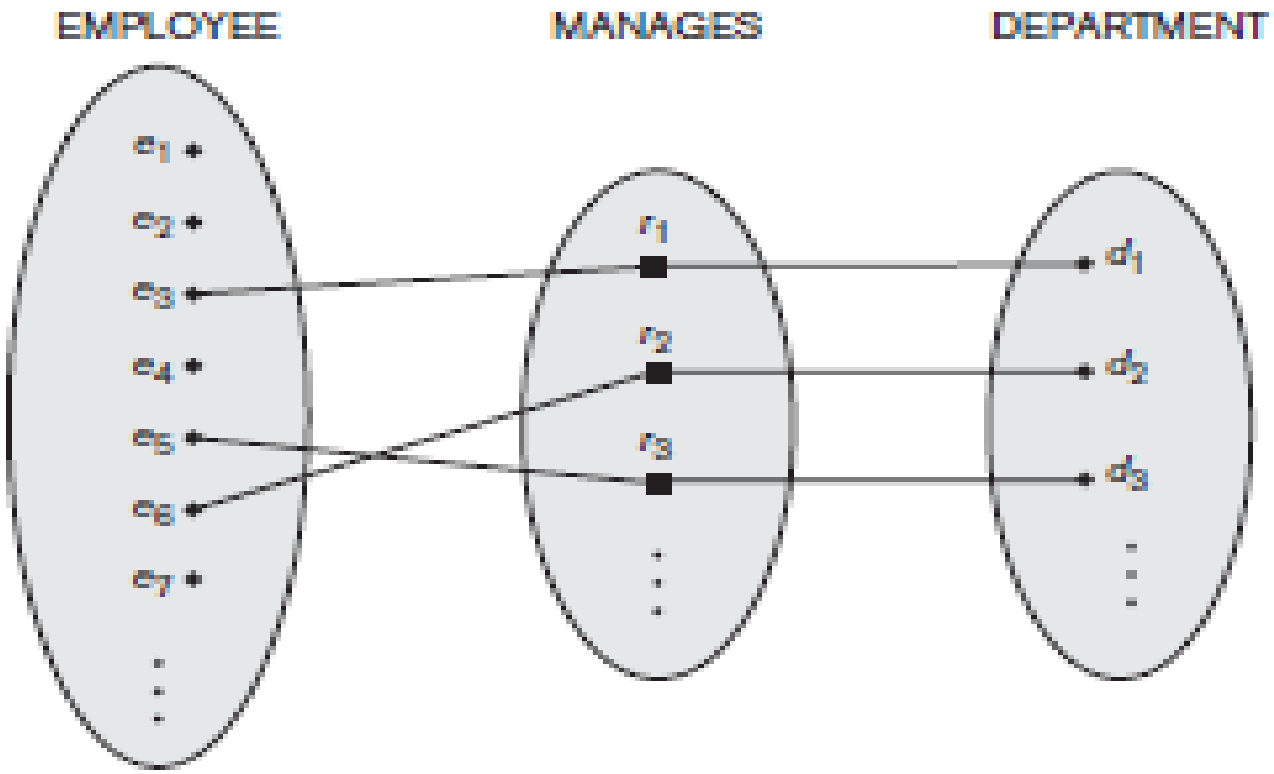


Cardinality Constraints on Relationship Types

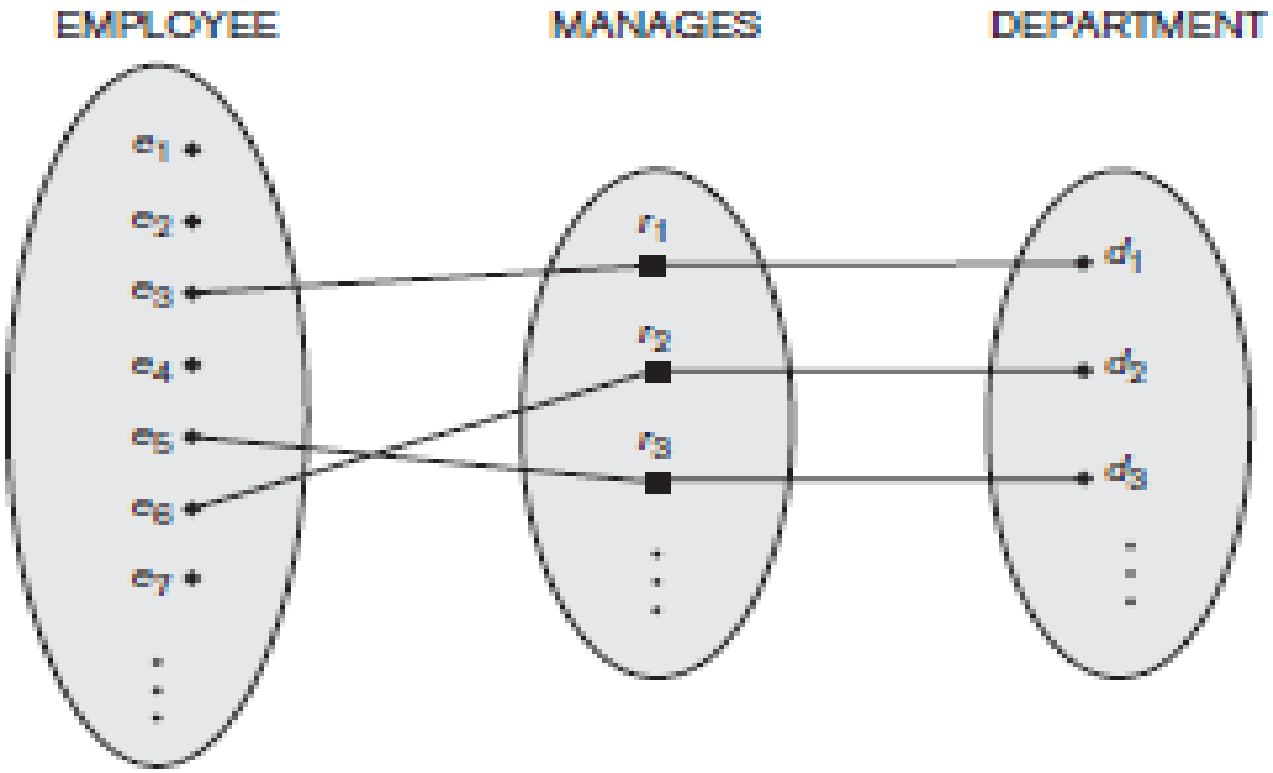
- The number of entities in an entity type participated in a relationship type is called **Cardinality Constraints** (also known as ratio constraints).
- Cardinality Constraints are of 2 types.
  - Maximum Cardinality
    - One-to-one
    - One-to-many
    - Many-to-many
  - Minimum Cardinality (also called participation or existence dependency constraints)
    - zero (partial participation, optional participation, not existence-dependent)
    - one or more (total participation, mandatory, existence-dependent)

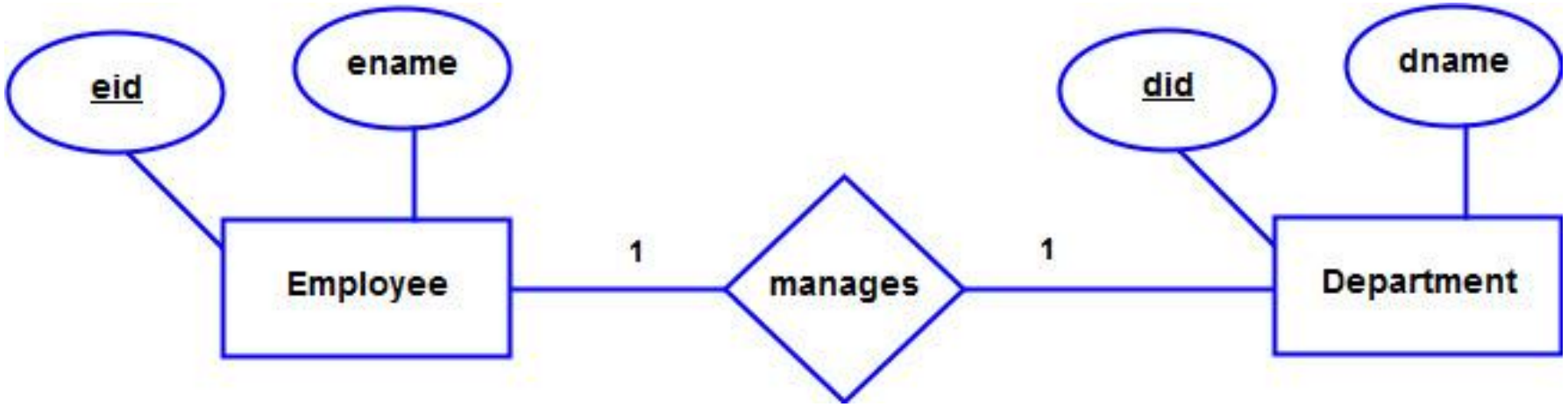
Maximum Cardinality Constraints

- Maximum Cardinality defines the maximum number of entities in an entity type participated in a relationship type.

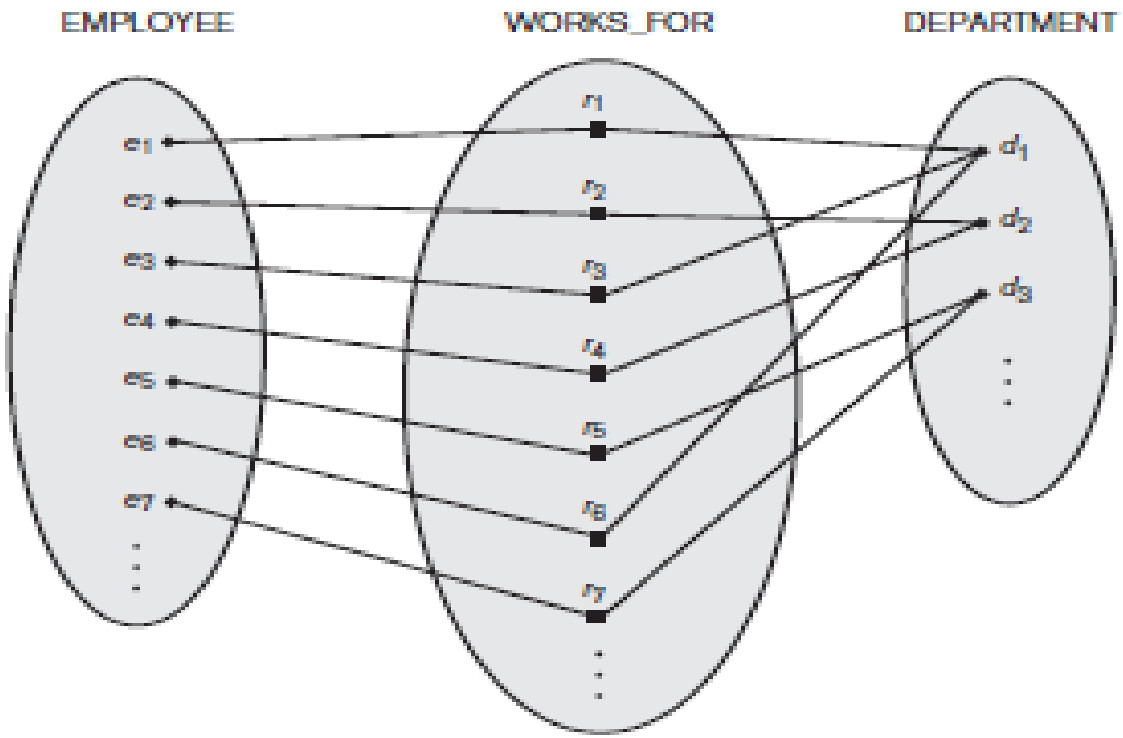


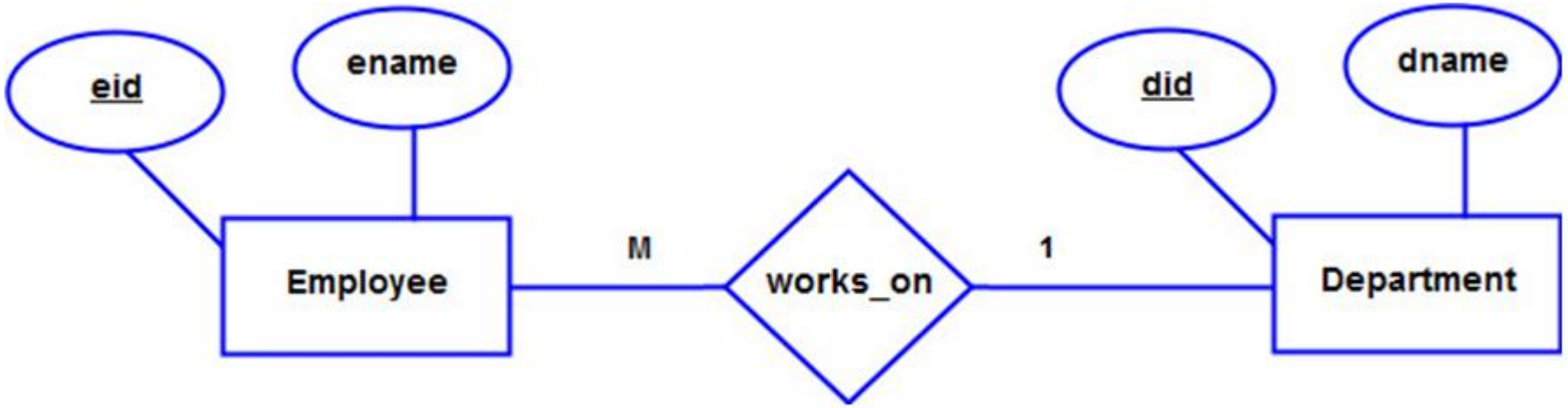
One-to-one


EMPLOYEE
MANAGES
DEPARTMENT
e_1
e_2
e_3
e_4
e_5
e_6
e_7
r_1
r_2
r_3
d_1
d_2
d_3

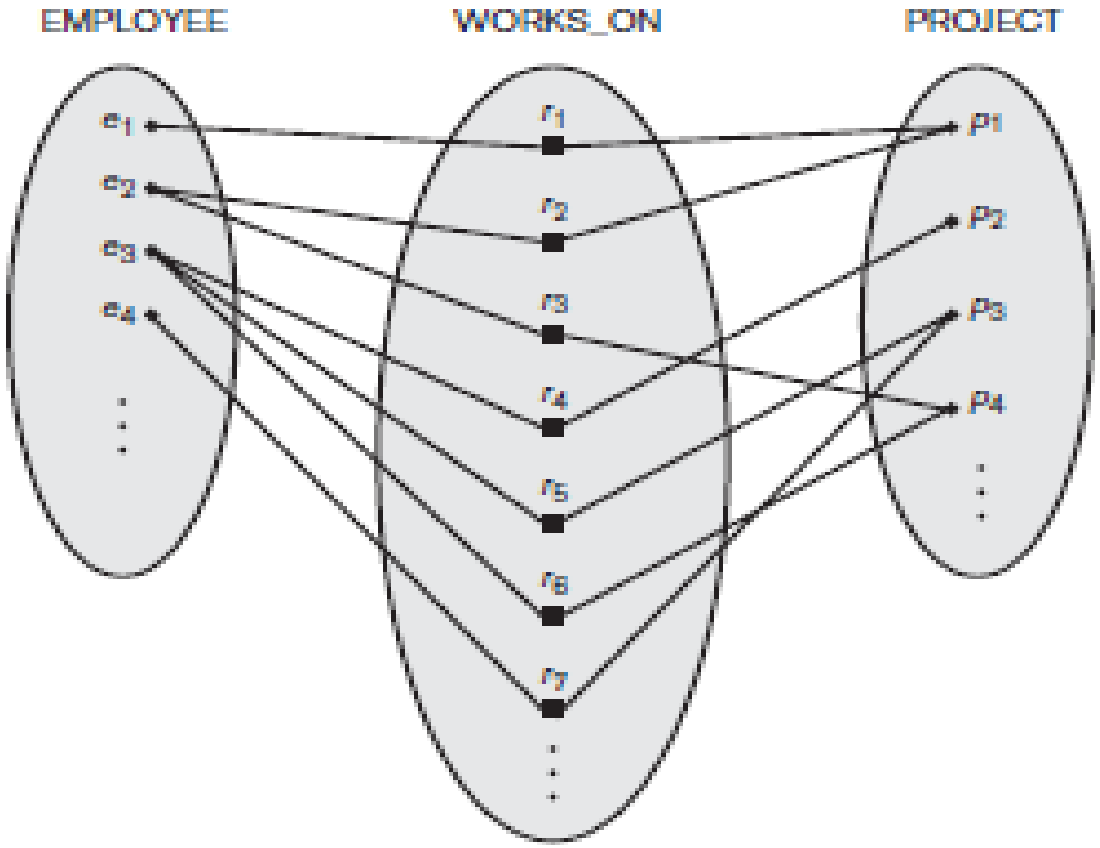

eid
ename
did
dname
Employee
1
manages
1
Department

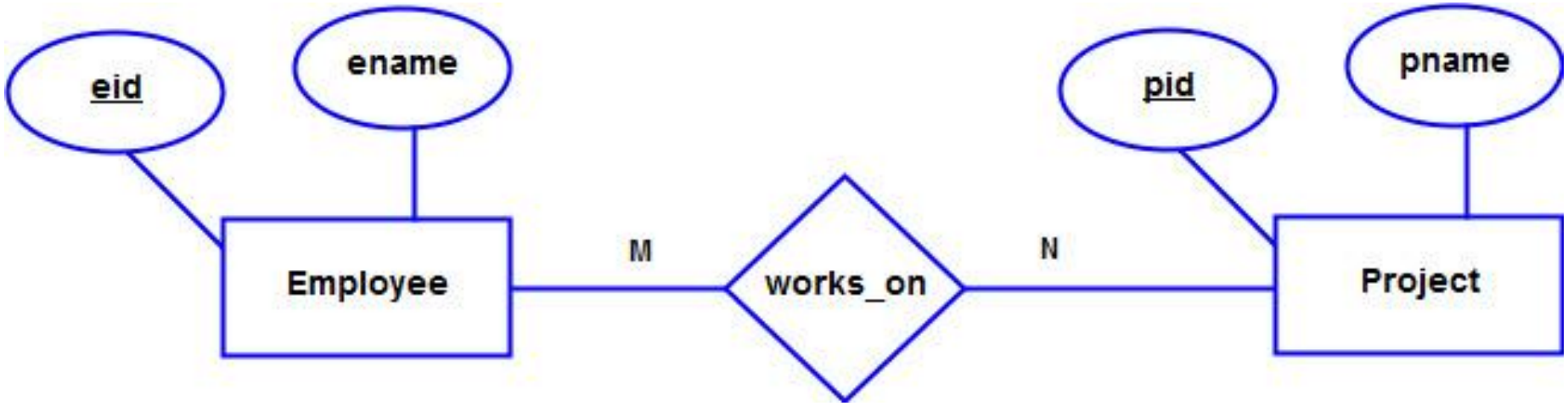
One-to-many / Many-to-one





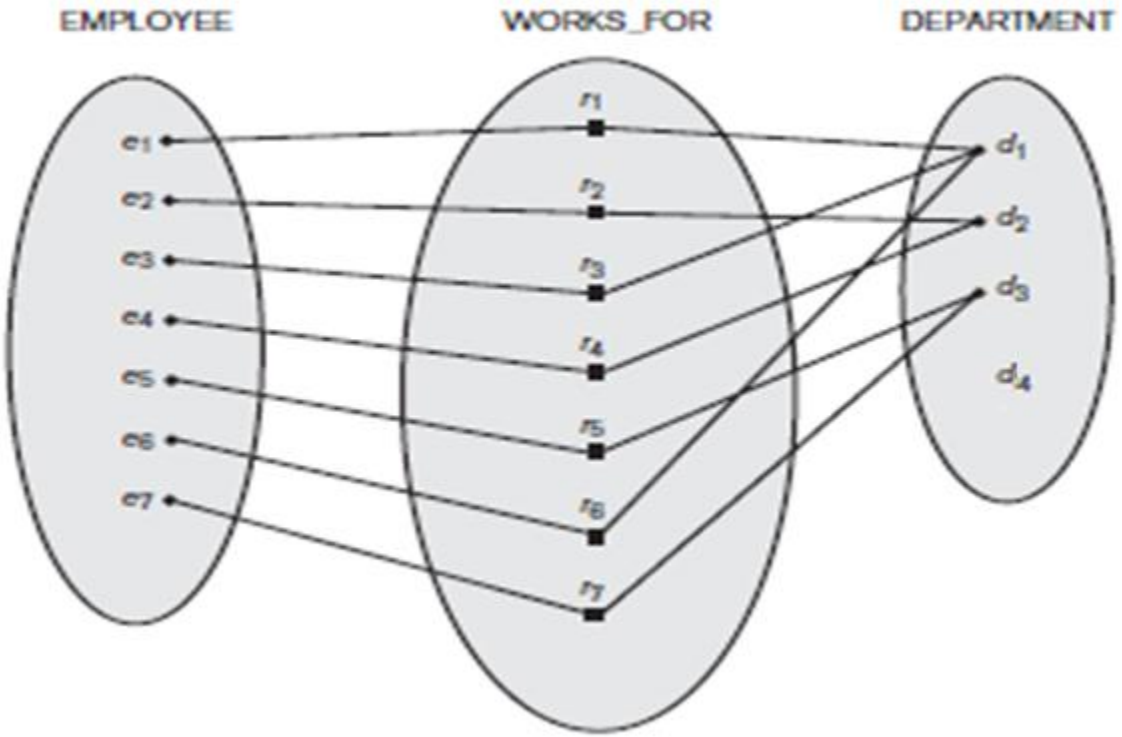
Many-to-many


EMPLOYEE
WORKS_ON
PROJECT
e1
e2
e3
e4
r1
r2
r3
r4
r5
r6
r7
p1
p2
p3
p4


eid
ename
pid
pname
Employee
M
works_on
N
Project

Minimum Cardinality Constraints

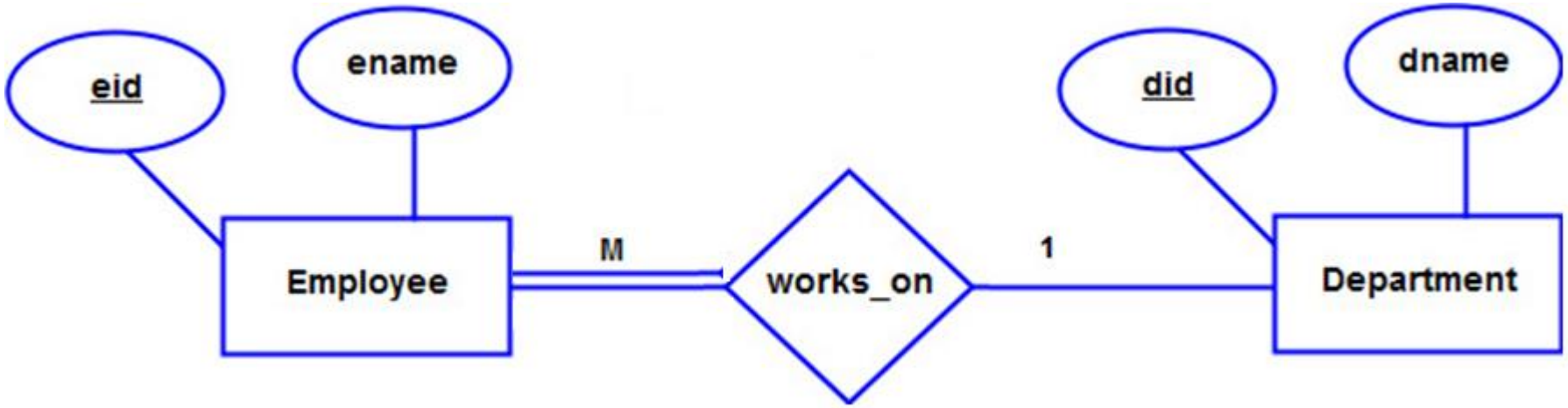
- Minimum Cardinality defines the minimum number of entities in an entity type participated in a relationship type.



Total and Partial participation

- Every entities in an entity type participated in a relationship type is called **total participation** of that entity type in that relationship type.

- Some of the entities in an entity type participated in a relationship type is called **partial participation** of that entity type in that relationship type.



Weak Entity type

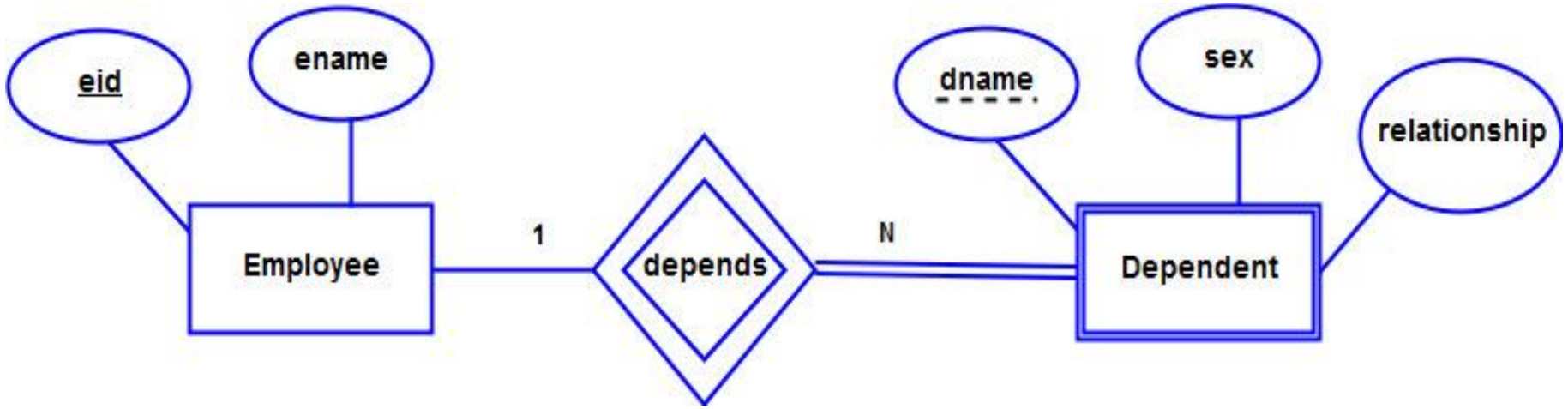
- An entity that does not have a key attribute
- A weak entity must participate in an **identifying relationship** type with an owner or identifying entity type
- Entities are identified by the combination of:
  - A partial key of the weak entity type
  - The particular entity they are related to in the identifying entity type

Example:

DEPENDENT is a weak entity type with EMPLOYEE as its identifying entity type via the identifying relationship type DEPENDENT_OF

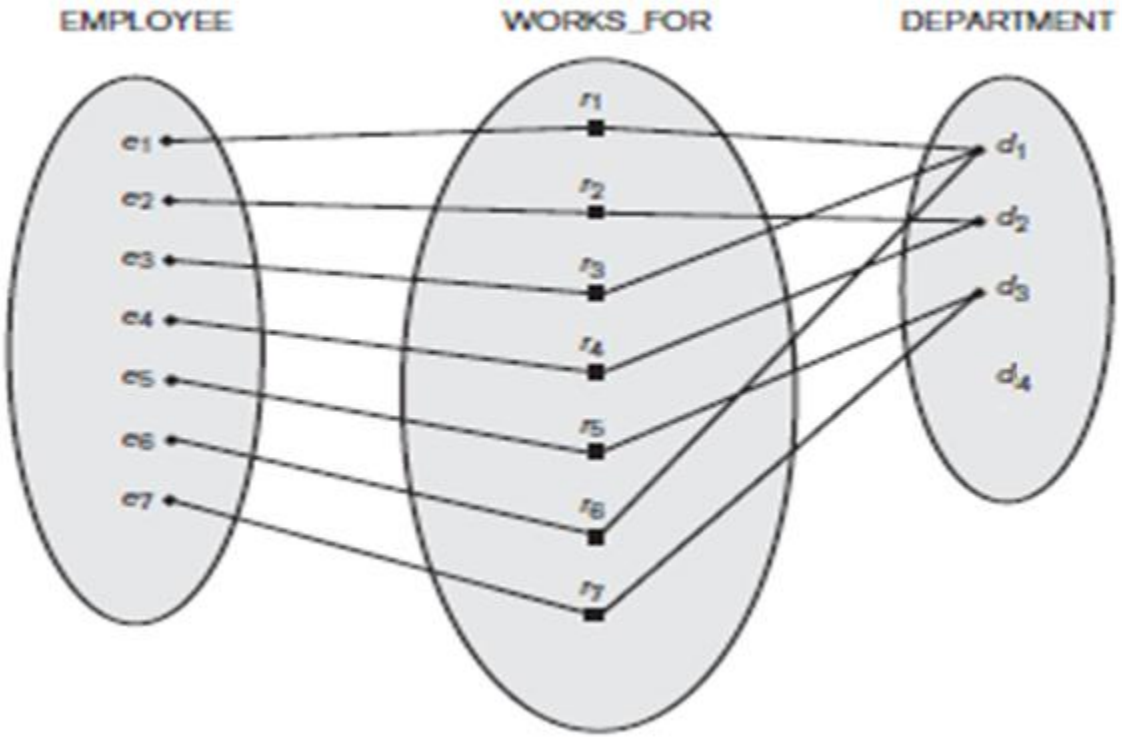
Weak Entity type (cont)

- A weak entity type is always in total participation with identifying relationship type.
- Cardinality constraint from weak entity type to identifying entity type is always N:1



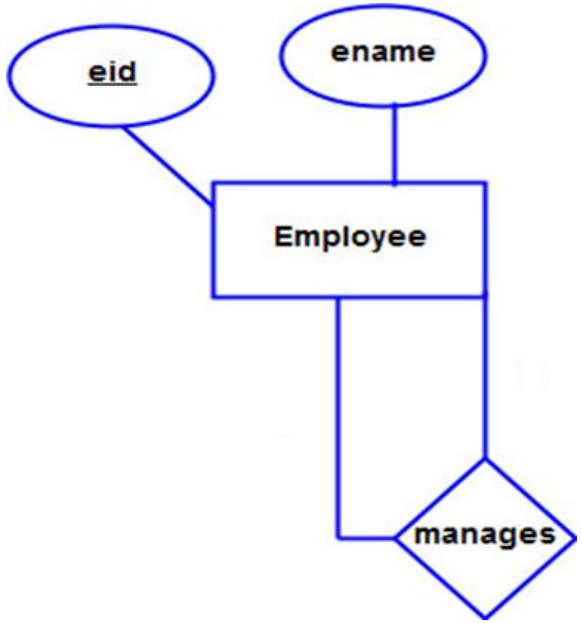
Role played by entity types in a relationship type

- Every entity type can play a particular role in a relationship type.



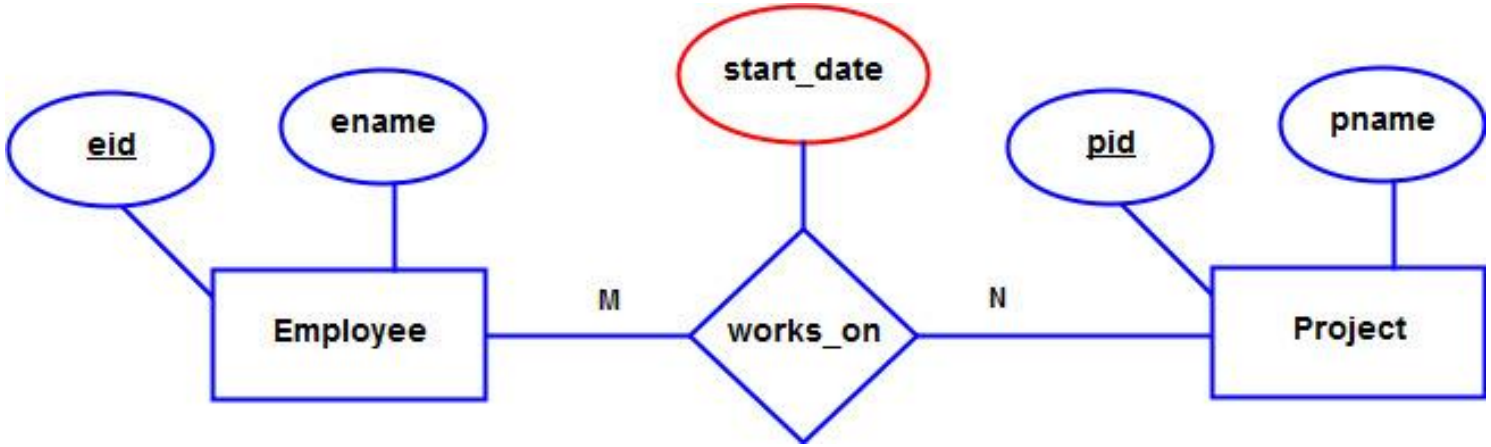
Recursive relationship type

- An entity type can play more than one role in a relationship type
- Eg: one EMPLOYEE (in the role of supervisee) to another EMPLOYEE (in the role of supervisor).



Attributes on Relationship type

- An EMPLOYEE can work on multiple project and a PROJECT can have more than one employees.
- Here, an EMPLOYEE started working on a particular PROJECT using start_date attribute.
- Hence, start_date belongs to relationship type WORKS_ON



References

- R. Elmasri and S.B. Navathe, “ Fundamentals of Database Systems,” Ed. 3., Addison Wesley, 2000, Chapters 3, 4.