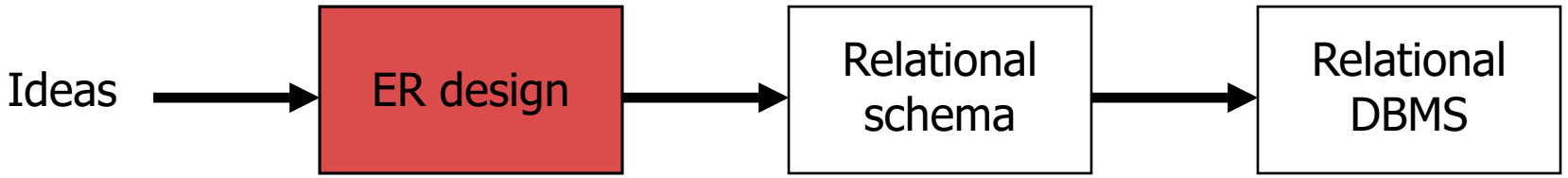
# Chapter 2

# Entity-Relationship Data Model

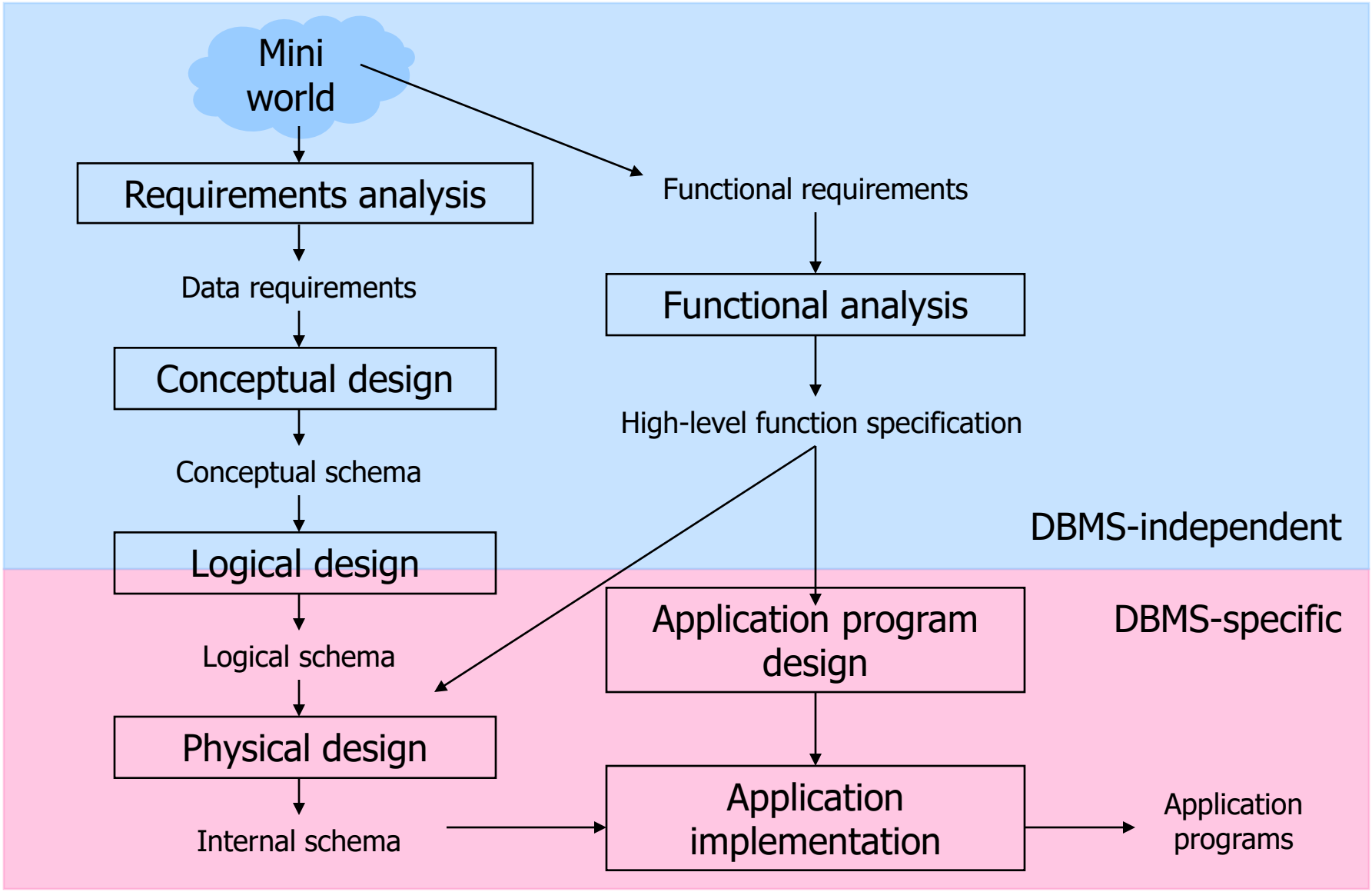
# Content

- Process of database design
- Example
- ER model
- ER model design

# Process of database design



# Process of database design



# Example

- “Company” database keeps track of a company’s employees, departments, and projects
  - (1) The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations.
  - (2) A department controls a number of projects, each of which has a unique name, a unique number, and a single location.

# Example

- “Company” database keeps track of a company’s employees, departments, and projects
  - (3) We store each employee’s name, SSN, address, salary, sex, and birth date. An employee is assigned to one department but may work on several projects, which are not necessarily controlled by the same department. We keep track of the number of hours that an employee works on each project. We also keep track of the direct supervisor of each employee.
  - (4) We want to keep track of the dependents of each employee for insurance purpose. We keep each dependent’s first name, sex, birth date, and relationship to the employee.

# Content

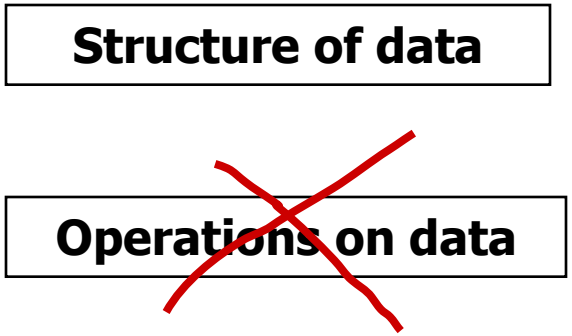
- Process of database design
- Example
- **ER model**
  - Entity
  - Attribute
  - Relationship
  - ER schema
  - Keys in ER model
  - Weak entity
- ER model design

# Entity-Relationship model

- Is used to design a DB at the conceptual level
- Abstract representation of the structure of a DB
  - Is represented graphically
- An entity-relationship model
  - Entity sets
  - Attributes
  - Relationships

# Entity sets

- An entity is an abstract object in the real world
  - Physical existence (person, car, house…)
  - Conceptual existence (company, job, university course...)
- A collection of similar entities forms an entity set
- Note
  - Entity
  - Object
  - Entity set
  - Class of objects



# Entity sets

- Example “Company” database
  - An employee is an entity
  - The set of all employees constitutes an entity set

  - Projects are entities
  - The set of projects is an entity set

  - A department is another kind of entity
  - The set of departments is an entity set

# Attributes

- The particular properties of the entities
- Example
  - An employee entity may be described by
    - Name
    - Age
    - Address …
- Assume that attributes are *atomic values*
  - Strings
  - Integers
  - Reals

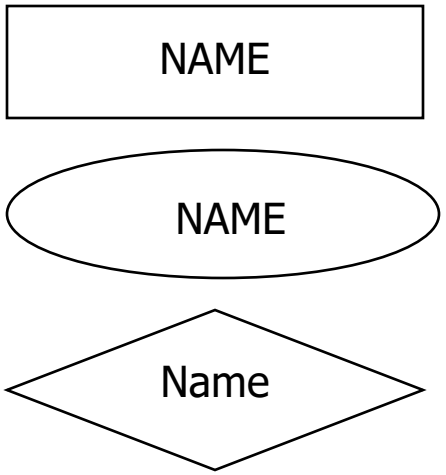
# Relationships

- The connections among two or more entity sets

- Example
  - Relationships between entity sets EMPLOYEE and DEPARTMENT
    - Each employee works for a department
    - Each department has an employee who manages that department

# ER model

- A graph representing entity sets, attributes, and relationships
  - Nodes

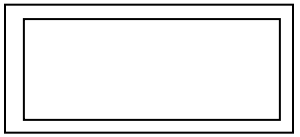
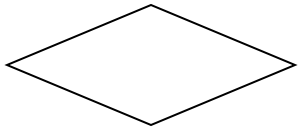
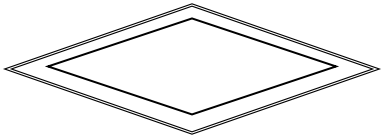


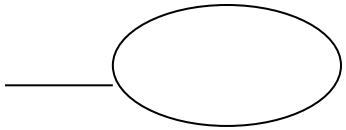
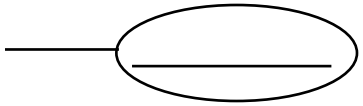
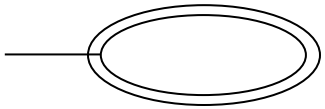
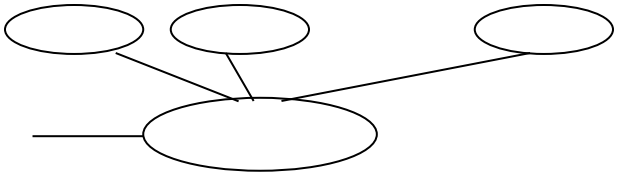

**Entity sets**

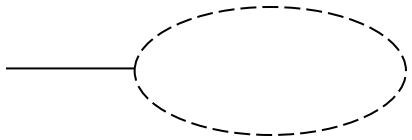
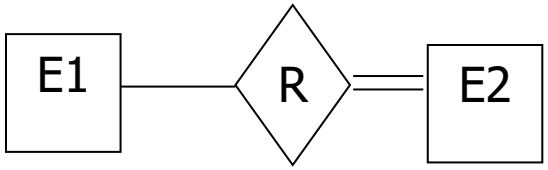
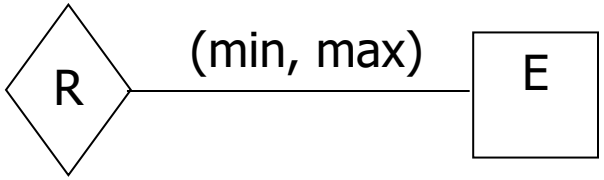
**Attributes**

**Relationships**

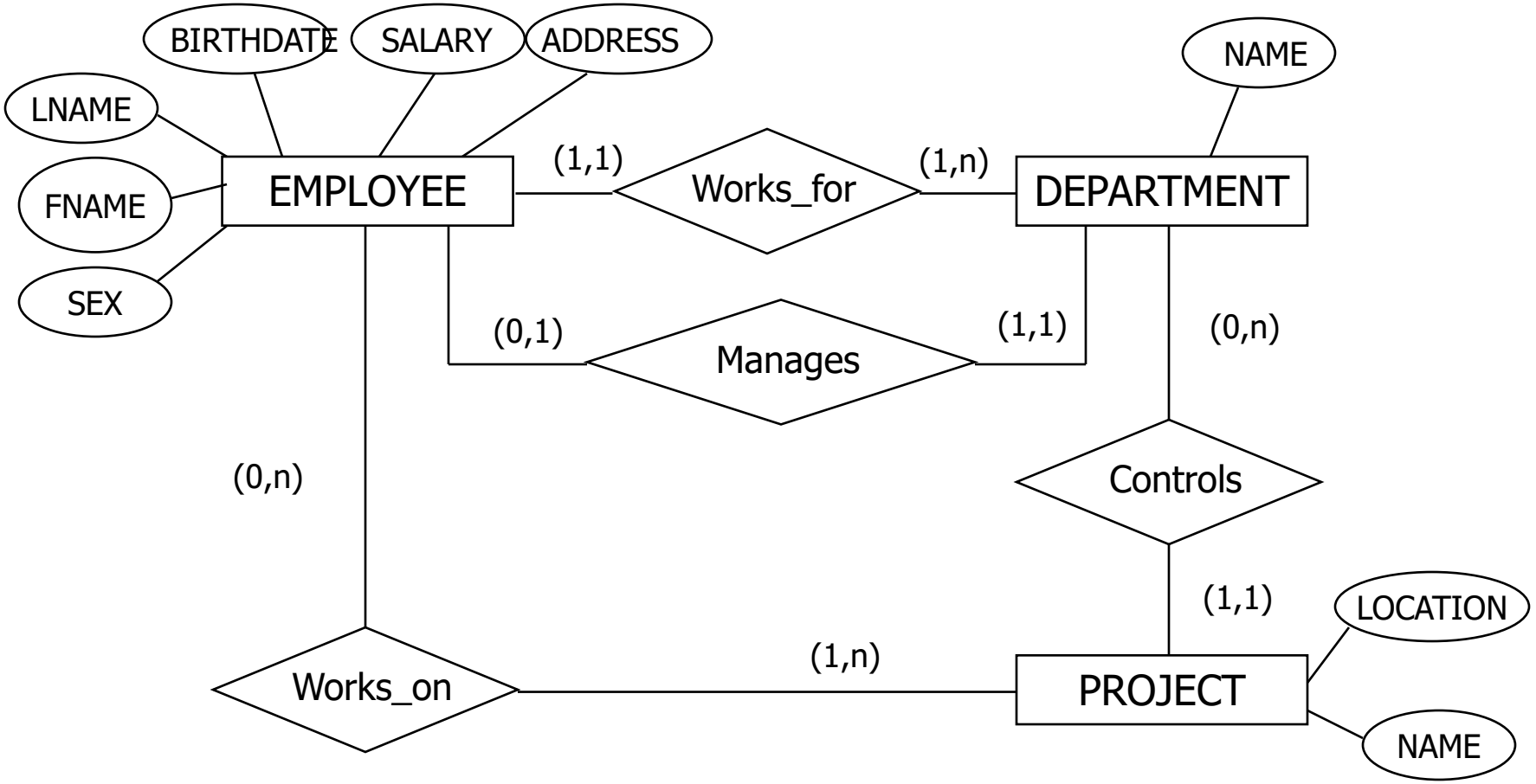
  - Edges connect
    - An entity set and its attributes
    - A relationship and its entity sets

| Symbol | Meaning |
| --- | --- |
|  | Entity |
|  | Weak entity |
|  | Relationship |
|  | Identifying relationship |

| Symbol | Meaning |
| --- | --- |
|  | Attribute |
|  | Key Attribute |
|  | Multivalued Attribute |
|  | Composite Attribute |

| Symbol | Meaning |
| --- | --- |
|  | Derived Attribute |
|  | Total Participation of E2 in R |
|  | E tham gia vaøo R theo tæ leä laø min:max |

# Example of an ER model



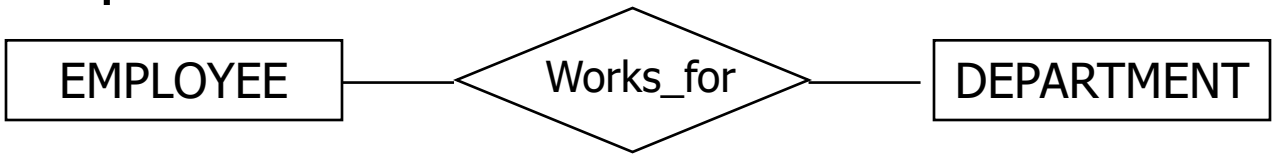
# Instances of an ER model

- A DB described by an ER model will contain particular data, which is called the DB instance
  - Each entity set, the instance will have a particular finite set of entities
    - Entity set EMPLOYEE has entities such as $E_1$, $E_2$, …, $E_n$
  - Each of entity has particular values for each attribute
    - $E_1$ has FName= *"Tung"*, BirthDate= *"08/12/1955"*, Sex= *"Nam"*
    - $E_2$ has Fname= *"Hang"*, BirthDate= *"07/19/1966"*, Sex= *"Nu"*
- Note
  - Do not store ER data directly in a DB
    - Is abstract only
    - Help us to think about the design before we convert to "relations"

# Relationship – Instance

- DB instance includes specific choices of the relationships of the diagram
  - A relationship R connects n entity sets $E_1$, $E_2$, …, $E_n$
  - Instance of R consists of a finite set of lists ($e_1$, $e_2$, …, $e_n$)
  - Each $e_i$ is chosen from the entities that are in $E_i$

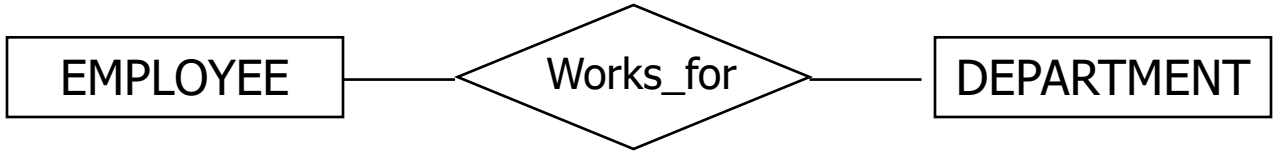
- Example




| EMPLOYEE | DEPARTMENT | |
|---|---|---|
| Tung | Nghien cuu | (Tung, Nghien cuu) |
| Hang | Dieu hanh | (Hang, Dieu hanh) |
| Vinh | Quan ly | (Vinh, Quan ly) |

# Relationship degree

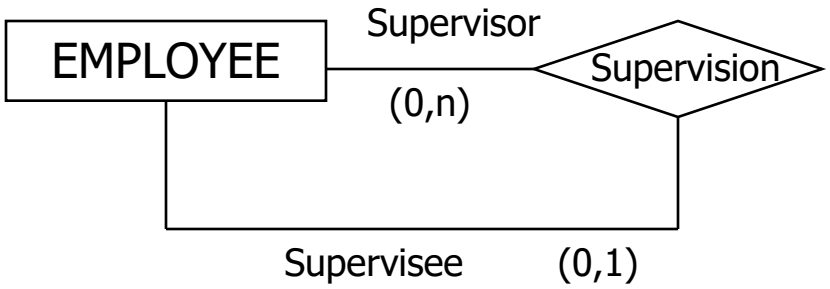
- The degree of a relationship type is the number of participating entity types.




  - Works-for is of degree two.
- A relationship type of degree two is called binary, and one of degree three is called ternary.

# Relationship – Role

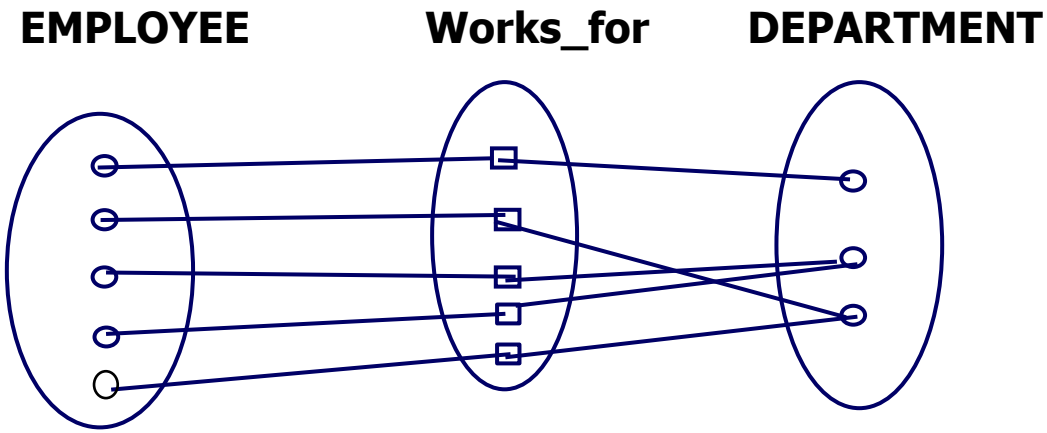
- One entity set appears two or more times in a single relationship
  - Draw many lines from the relationship to the entity set
  - Each line represents a different role that the entity set plays in the relationship




  - Supervision is called recursive relationship or self-referencing relationship.

# Constraints on binary relationship types

1. Cardinality ratios
   - Specifies the maximum number of relationship instances that an entity can participate in.

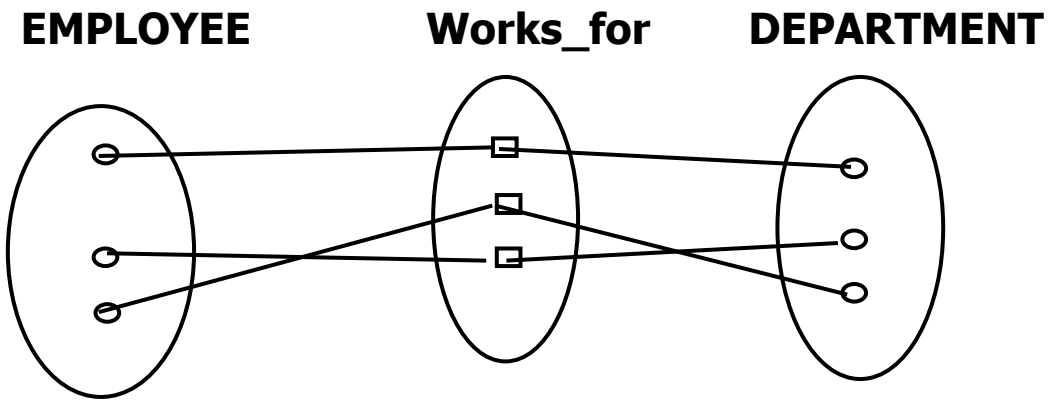



   - Each department can be related to any number of employees (n), but an employee can be related to (works_for) at most one department (1).

# Constraints on binary relationship types

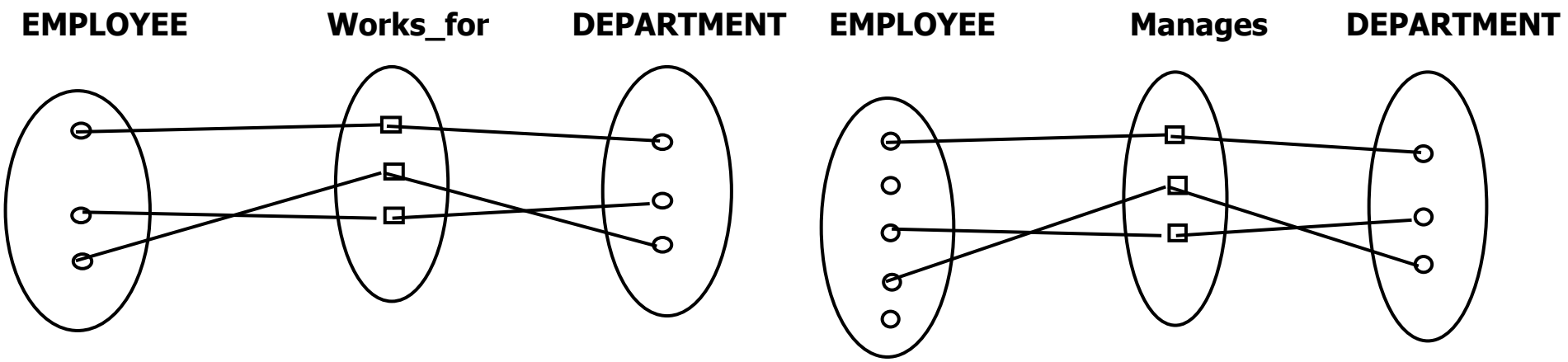
2. Participation constraint

- Specifies the minimum number of relationship instances that each entity can participate in.




- Every employee must work for a department (1).

# Structural contraints



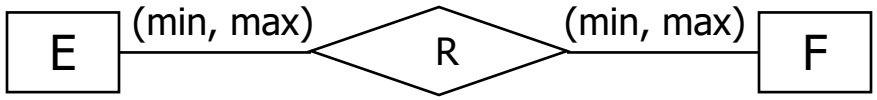

- Total participation: participation of EMPLOYEE in “Works_for”.
- Partial participation: participation of EMPLOYEE in “Manages”.
- Structural constraints = Total participation & partial participation constraints

# Relationship – Multiplicity

- A binary relationship R connecting entity sets E and F, then
  - Many-One from E to F
  - One-One from E to F
  - Many-Many from E to F

# Relationship – Multiplicity

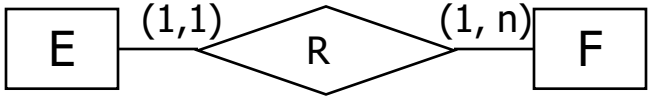
- (min,max) specifies the minimum or maximum number that each entity e ∈ E can participate in R

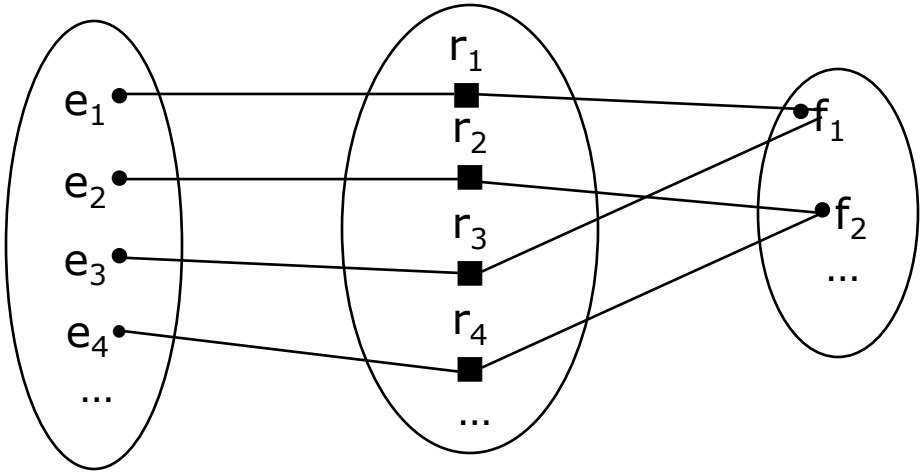



  - (0,1) – zero or 1
  - (1,1) – at least 1 and at most 1
  - (0,n) – zero or many
  - (1,n) – 1 or many

# Relationship – Multiplicity

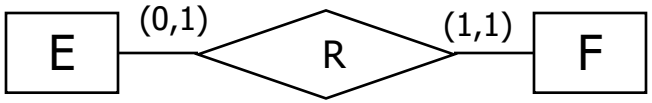
- One-One from E to F ~ One-Many from F to E
  - Each member of E can be connected at most one member of F
  - Each member in F can be connected to many members of E

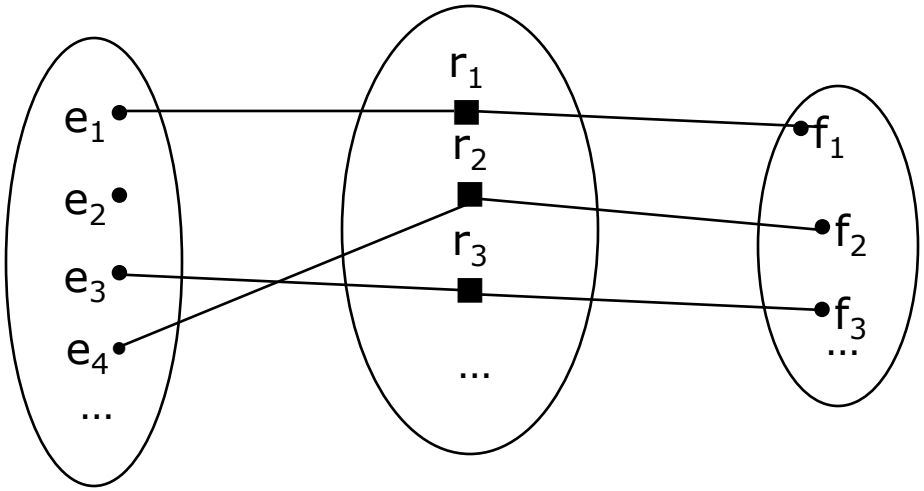


# Relationship – Multiplicity

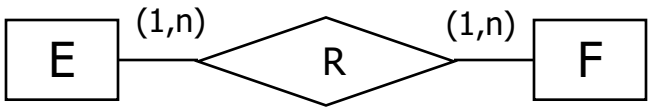
- One-One from E to F
  - A member of either entity set can be connected to at most one entity of the other set

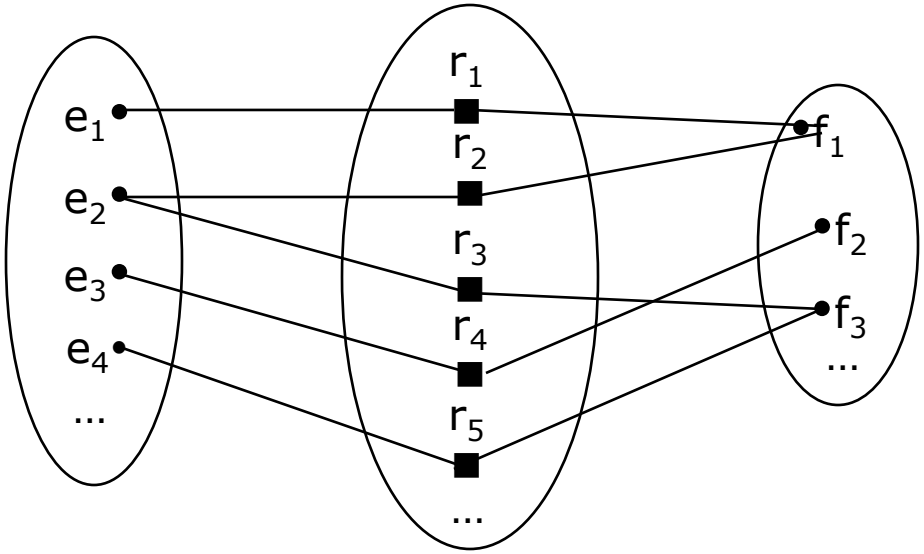


# Relationship – Multiplicity

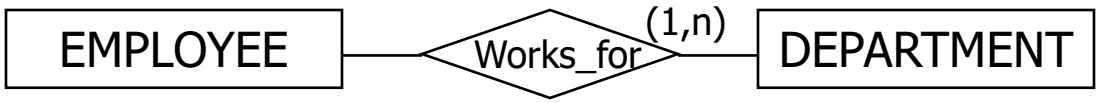
- Many-Many from E to F
  - A member of either entity set can be connected to many entities of the other set



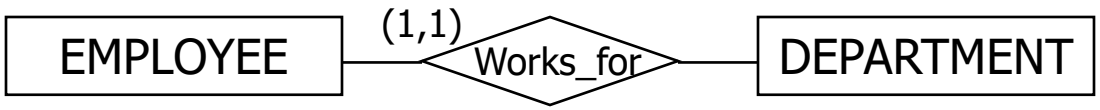



# Relationship – Multiplicity

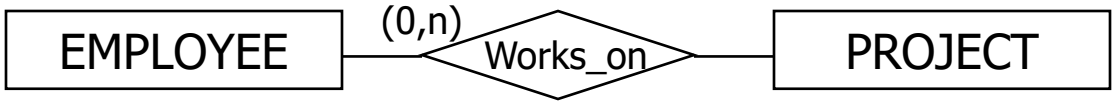
- Example
  - A department has many employees




  - An employee works for a department



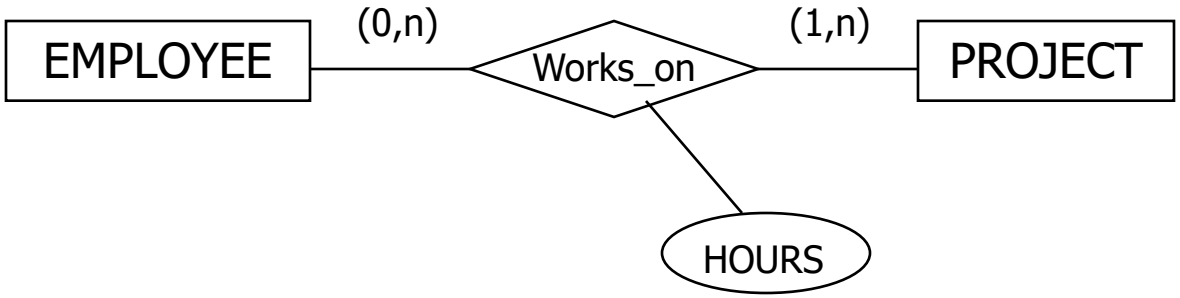

  - An employee can work on many projects or do not work on any projects




  - An employee can manage a certain department

# Attributes on relationships

- Attributes on a relationship describe properties for that relationship
- These attributes cannot associate with entity sets that the relationship connects

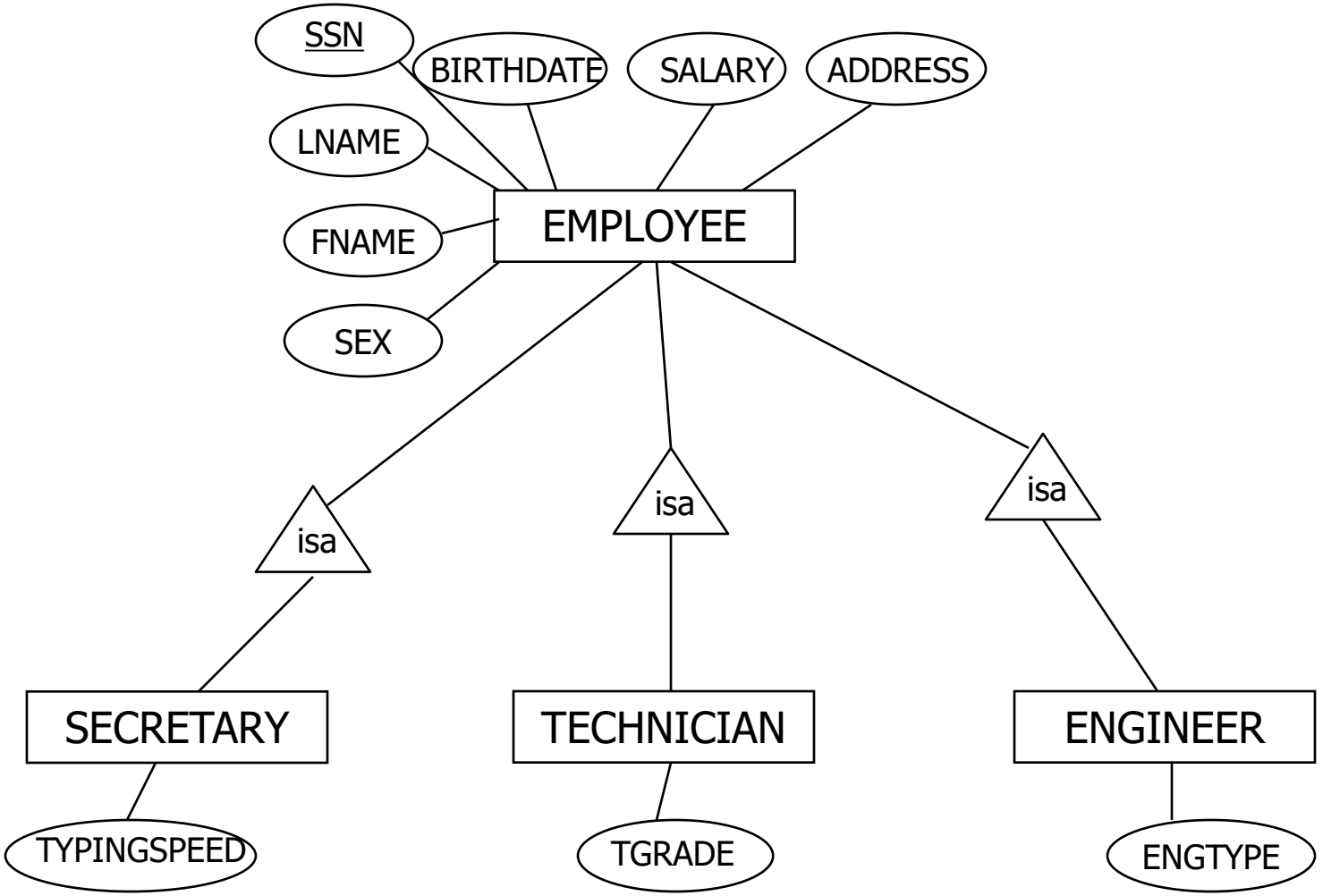


# Relationship “isa”

- An entity set contains certain entities that have special properties not associated with all member of the set

- Then, we define a special-case entity set
  - Subclass
  - Have its own special attributes and/or relationships
  - Connect to its superclass using the “isa” relationship
    - “an A is a B” = “isa” relationship from entity set A to entity set B

# Example


SSN
BIRTHDATE
SALARY
ADDRESS
LNAME
FNAME
SEX
EMPLOYEE
isa
isa
isa
SECRETARY
TECHNICIAN
ENGINEER
TYPINGSPEED
TGRADE
ENGTYPE

# Relationship “isa”

- Note
  - The “isa” relationship is the one-one relationship
  - One root entity set
  - Entity set E is a subclass, entity set F is superclass
    - Entity e ∈ E has whatever attributes that F has
    - Entity e ∈ E participates in whatever relationships that F participates in

# Key attributes

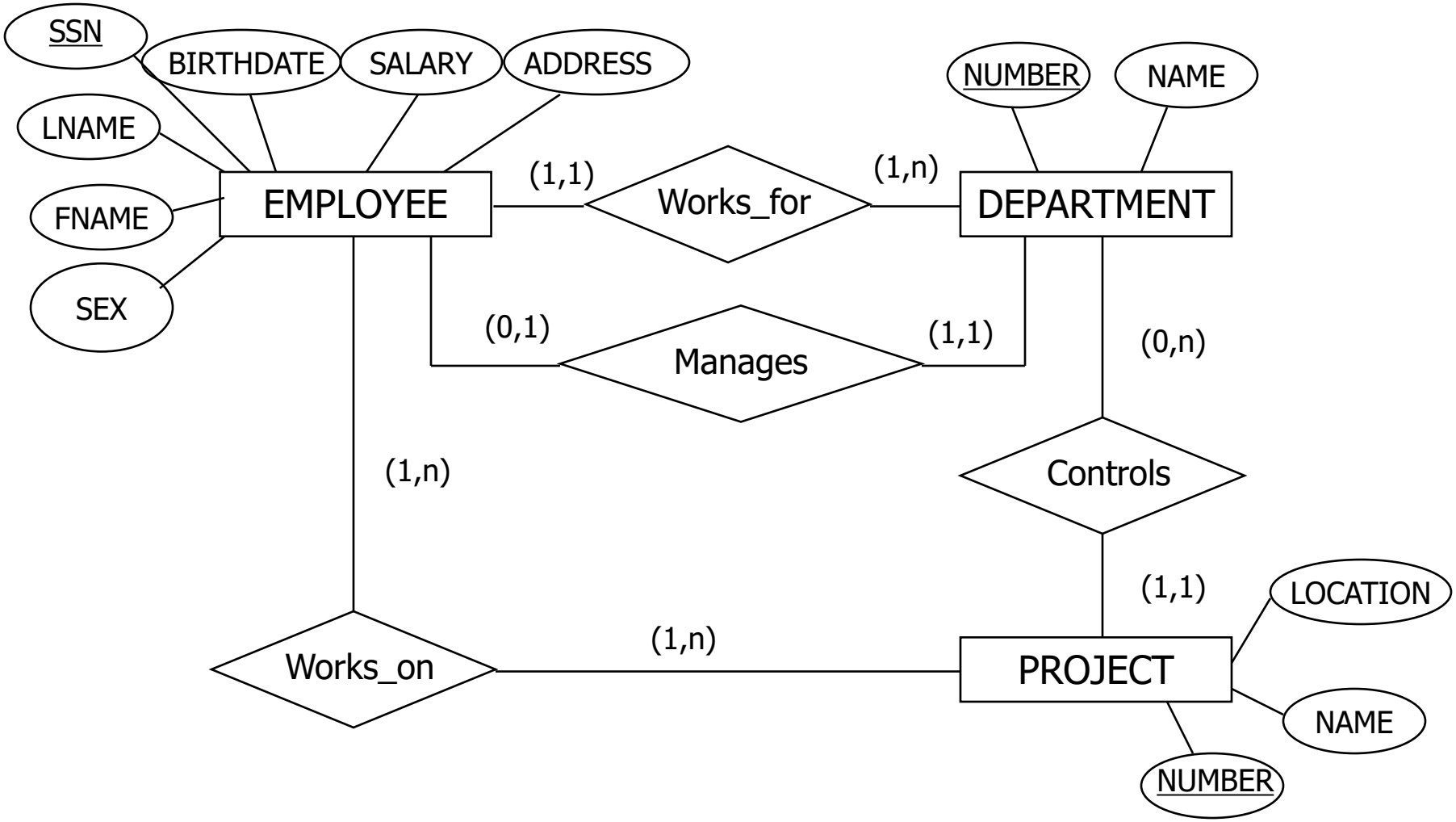
- All entities in an entity set are necessary to be distinguished

- A key K for an entity set E is one or more attributes such that
  - Given any two distinct entities $e_1$ and $e_2$ in E, $e_1$ and $e_2$ cannot have identical values for all of the attributes in the key K

# Key attributes

- Note
  - Every entity set must have a key
  - A key can consist of more than one attribute
  - There can be more than on possible key for an entity set
    - Pick on key as the “primary key” for that entity set
  - In an “isa” relationship, the root entity set needs its own key

# Example

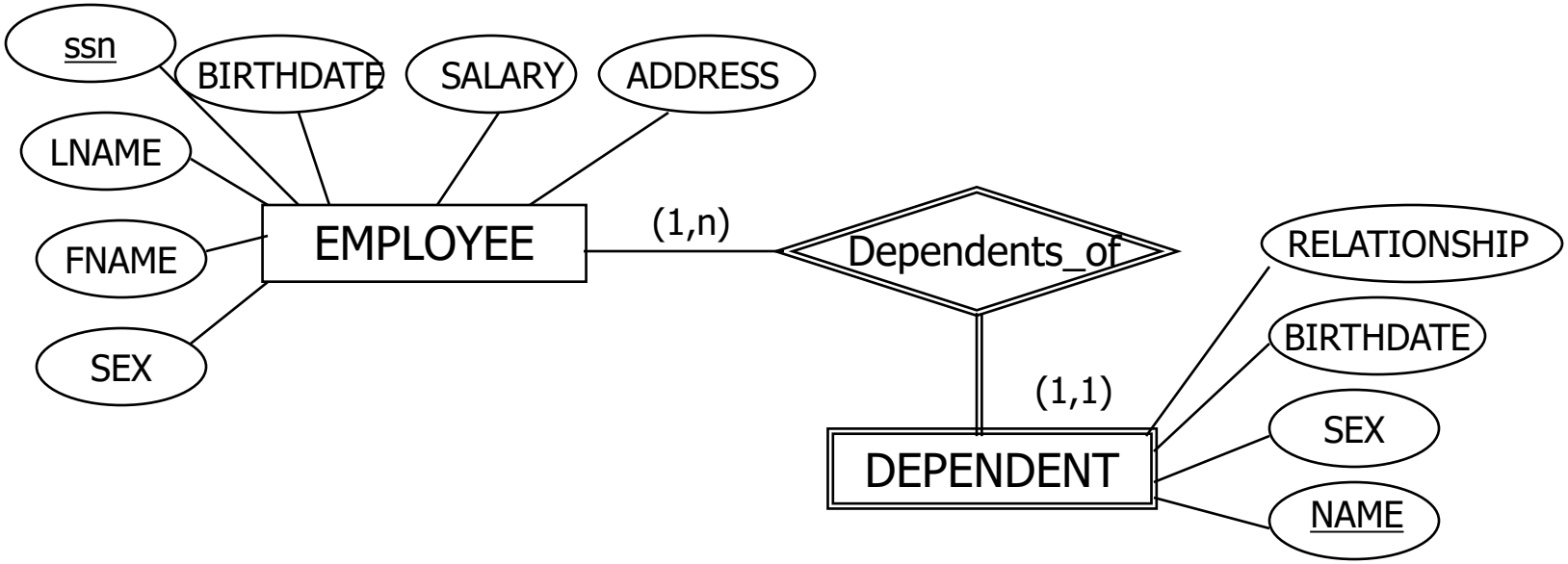

SSN
BIRTHDATE
SALARY
ADDRESS
LNAME
FNAME
SEX
EMPLOYEE
(1,1)
Works_for
(1,n)
NUMBER
NAME
DEPARTMENT
(0,1)
Manages
(1,1)
(0,n)
(1,n)
Controls
(1,1)
LOCATION
Works_on
(1,n)
PROJECT
NAME
NUMBER

# Weak entity set

- An entity set whose key is composed of attributes which belong to another entity set

- Sources of weak entity set
  - Entities of set E are sub-units of entities in set F
    - The names of E entities are not unique until we take into account the name of F entity to which the E entity is subordinate
  - Some entity sets have no attribute for their own
    - Their key is formed from attributes that are the key attributes for the entity sets they connect

# Example


ssn
BIRTHDATE
SALARY
ADDRESS
LNAME
FNAME
SEX
EMPLOYEE
(1,n)
Dependents_of
(1,1)
DEPENDENT
RELATIONSHIP
BIRTHDATE
SEX
NAME

# Content

- Process of database design
- Example
- E/R model
- **E/R model design**
  - Steps of design
  - Design principles
  - ER model of “Company”

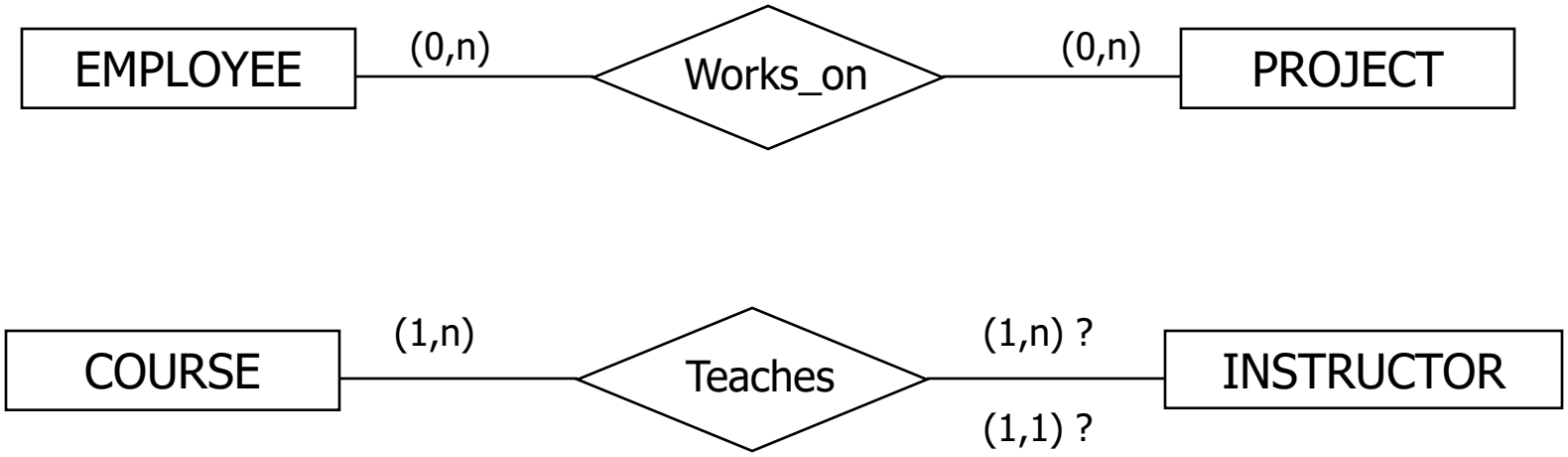
# Steps of design

- (1) Determine entity sets
- (2) Determine relationships among entity sets
- (3) Determine attributes and connect them to entity sets and relationships
- (4) Specify the domain for attributes
- (5) Decide key attributes
- (6) Specify (min, max) of relationships

# Design principles

- Faithfulness
  - Entity sets and their attributes should reflect reality
  - Whatever relationships are asserted should make sense

  - Examples



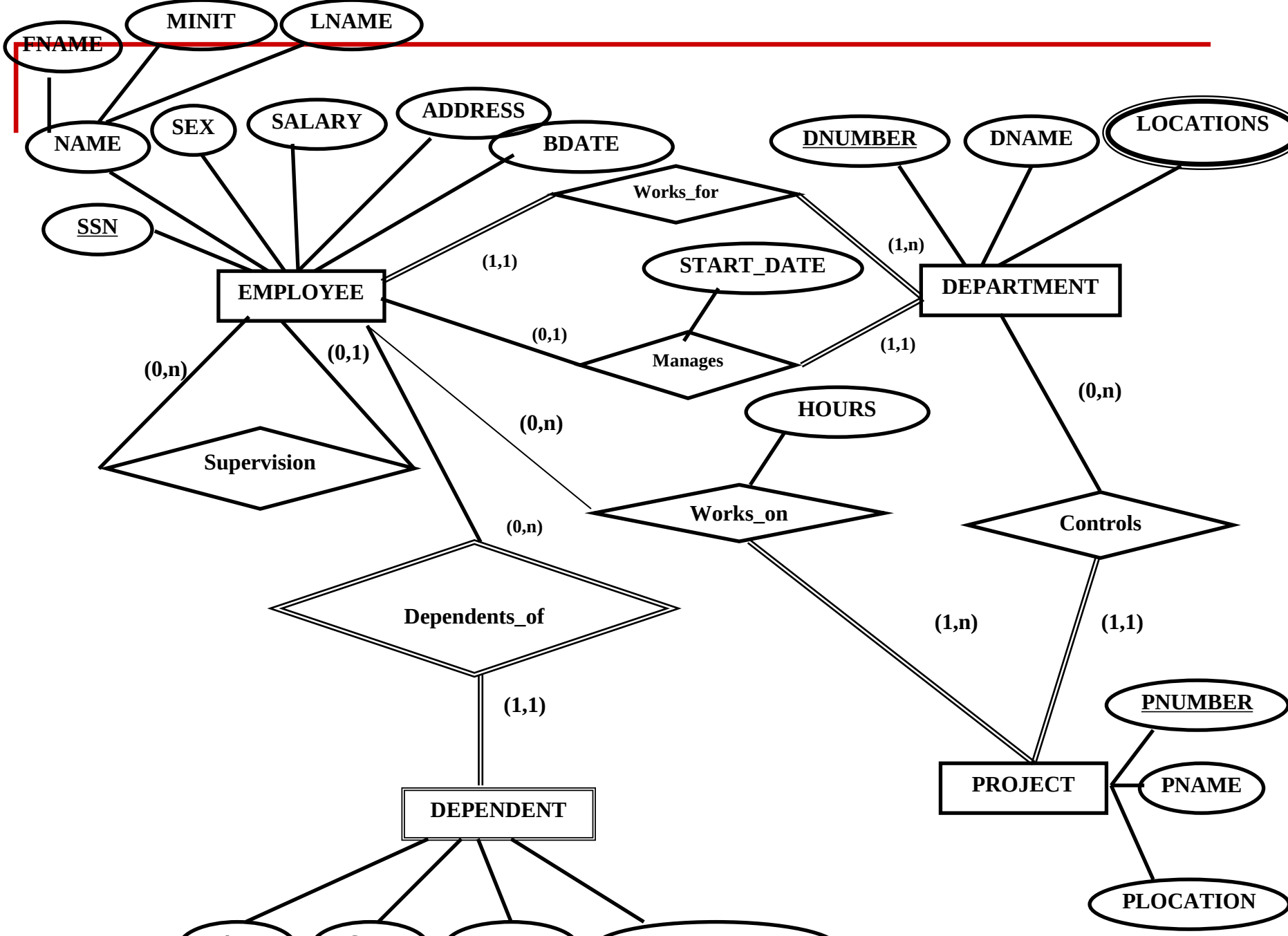
# Design principles

- Avoiding redundancy
- Simplicity
  - Avoid introducing more elements into your design than is absolutely necessary
- Choosing the right relationships
  - Adding to our design every possible relationship is not a good idea
    - Redundancy, modifying the DB becomes complex
- Picking the right kind of element
  - Using attributes
  - Using entity set/relationship combinations

# Content

- Process of database design
- Example
- E/R model
- E/R model design
- **ER model of “Company”**

- Draw a full ER model for the “Company” example


MINIT
LNAME
FNAME
NAME
SEX
SALARY
ADDRESS
BDATE
SSN
EMPLOYEE
Works_for
(1,1)
(1,n)
DNUMBER
DNAME
LOCATIONS
DEPARTMENT
START_DATE
(0,1)
Manages
(1,1)
(0,n)
(0,1)
(0,n)
Supervision
(0,n)
HOURS
Works_on
Controls
(0,n)
Dependents_of
(1,n)
(1,1)
(1,1)
PNUMBER
PROJECT
PNAME
DEPENDENT
PLOCATION
NAME
SEX
BIRTH_DATE
RELATIONSHIP