Entity-Relationship Model

Assoc. Prof. Tran Minh Quang

Faculty of Computer Science & Engineering, HCMUT

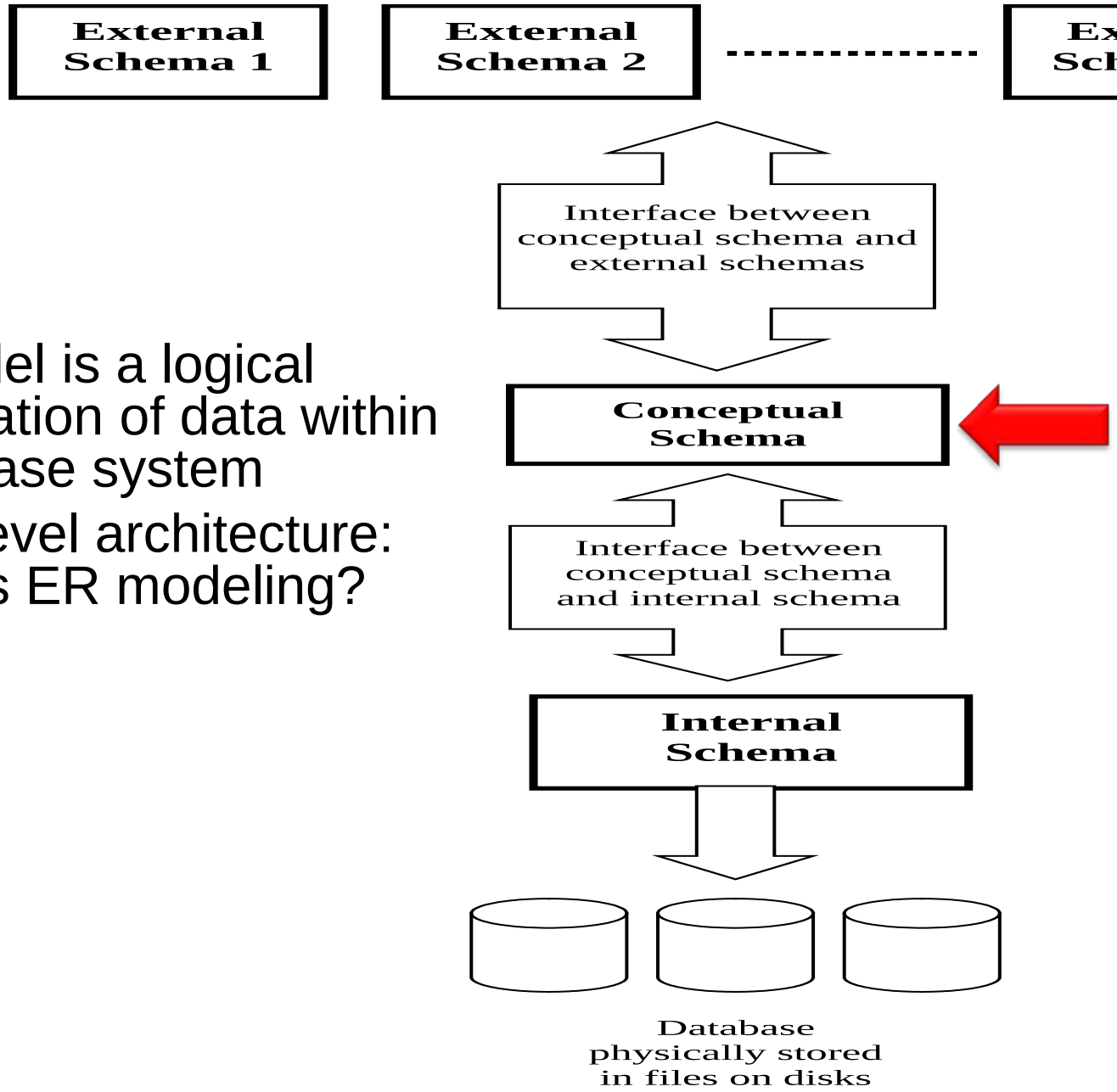
quangtran@hcmut.edu.vn

Outline

- What is ER Model? And Why?
- Overview of Database Design Process
- Example COMPANY Database
- ER Model Concepts
- ER Diagram
- Alternative Diagrammatic Notations
- Problems with ER Models
- Reading Suggestion:
  - [1]: Chapter 3
  - A. Badia: ”Entity-Relationship Modeling Revisited”, SIGMOD Record, 33(1), March 2004, 77-82

What is ER Model? And Why?

- ER model is a logical organisation of data within a database system
- Three-level architecture: where is ER modeling?



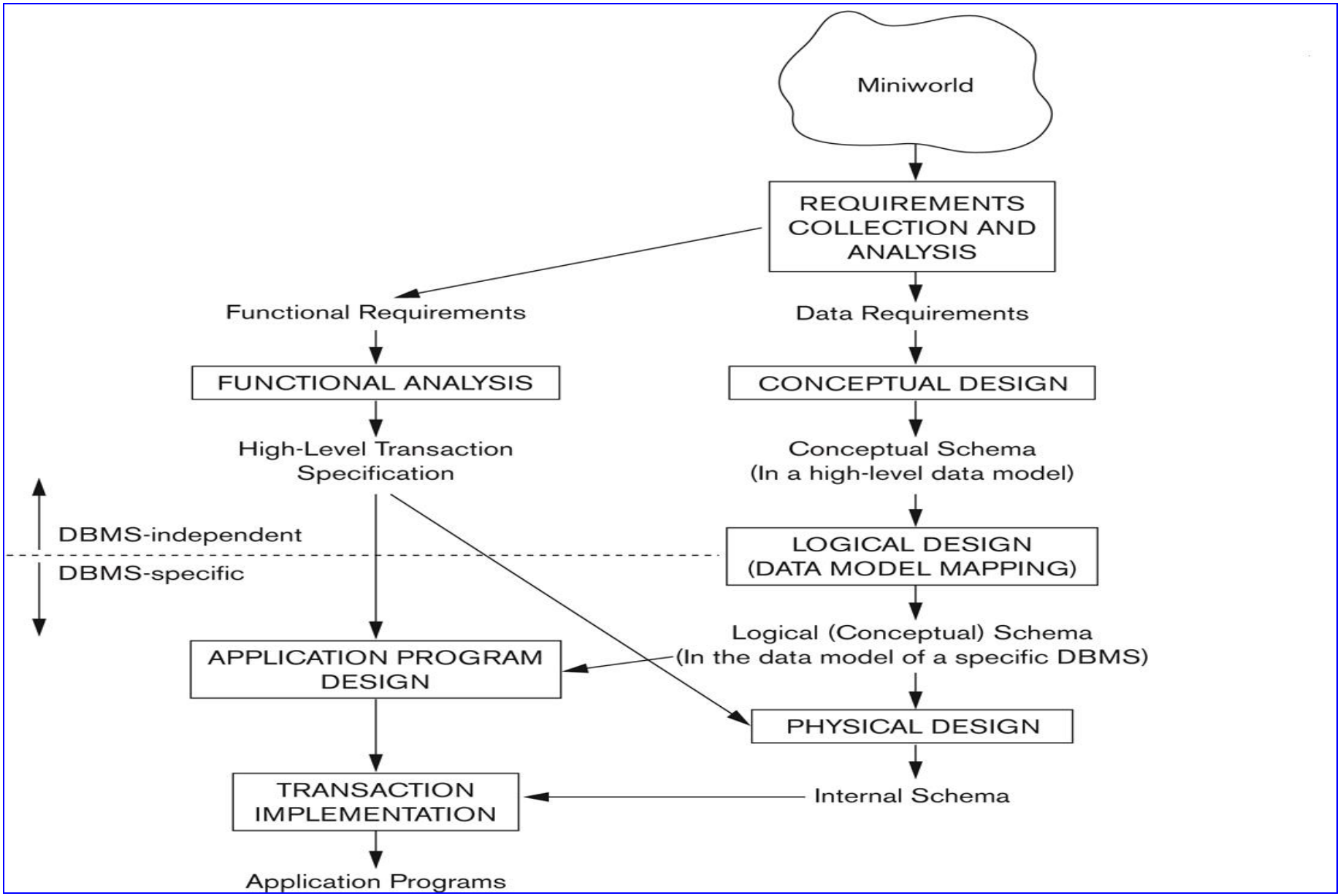
What is ER Model? And Why?

- ER model technique is based on relational data model
- Why use ER data modelling:
  - User requirements can be specified formally & unambiguously
  - The conceptual data model is independent of any particular DBMS
  - It does not involve any physical or implemental details
  - It can be easily understood by ordinary users
  - It provides an effective bridge between informal user requirements and logical database design and implementation

Overview of Database Design Process

- Two main activities:
  - Database design
  - Application design
- Focus in this lecture on database design:
  - To design the conceptual schema for a database application
- Application design focuses on the programs and interfaces that access the database
- Generally considered part of software engineering

Overview of Database Design Process



Outline

- What is ER Model? And Why?
- Overview of Database Design Process
- Example COMPANY Database
- ER Model Concepts
- ER Diagram
- Alternative Diagrammatic Notations
- Problems with ER Models
- Reading Suggestion:
  - [1]: Chapter 3
  - My homepage: A. Badia: ”Entity-Relationship Modeling Revisited”, SIGMOD Record, 33(1), March 2004, 77-82

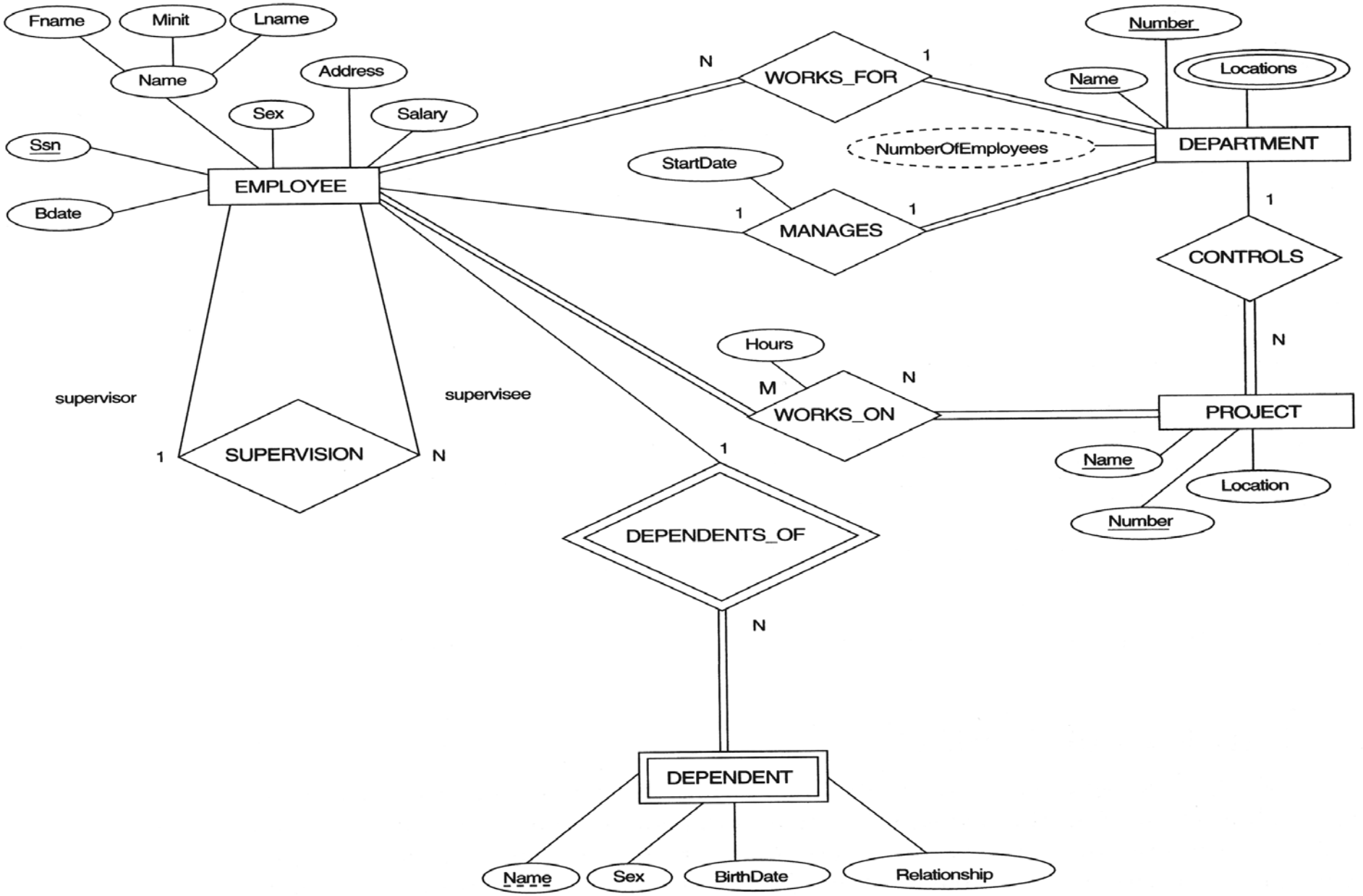
Example COMPANY Database

- Requirements of the Company (oversimplified for illustrative purposes)
  - The company is organized into DEPARTMENTs. Each department has a name, number and an employee who *manages* the department. We keep track of the start date of the department manager
  - Each department controls a number of PROJECTs. Each project has a name, number and is located at a single location

Example COMPANY Database

- Requirements of the Company (oversimplified for illustrative purposes)
  - We store each EMPLOYEE's social security number, address, salary, sex, and birthdate. Each employee works for one department but may work on several projects. We keep track of the number of hours per week that an employee currently works on each project. We also keep track of the direct supervisor of each employee
  - Each employee may have a number of DEPENDENTs. For each dependent, we keep track of their name, sex, birthdate, and relationship to employee

Example COMPANY Database



ER Model Concepts

- Entities and Attributes
  - Entities are specific objects or things in the mini-world that are represented in the database
    - →E.g., the EMPLOYEE John Smith, the Research DEPARTMENT, the ProductX PROJECT, etc.
  - Attributes are properties used to describe an entity
    - →E.g., an EMPLOYEE entity may have a Name, SSN, Address, Sex, BirthDate
  - A specific entity will have a value for each of its attributes
    - →E.g., a specific employee entity may have Name='John Smith', SSN='123456789', Address ='731, Fondren, Houston, TX', Sex='M', BirthDate='09-JAN-55‘
  - Each attribute has a *value set* (or data type) associated with it
    - →E.g., integer, string, subrange, enumerated type, etc.

ER Model Concepts

- 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. For example, Address (Apt#, House#, Street, City, State, ZipCode, Country) or Name (FirstName, MiddleName, LastName). Composition may form a hierarchy where some components are themselves composite
  - Multi-valued
    - →An entity may have multiple values for that attribute. For example, Color of a CAR or PreviousDegrees of a STUDENT. Denoted as {Color} or {PreviousDegrees}

ER Model Concepts

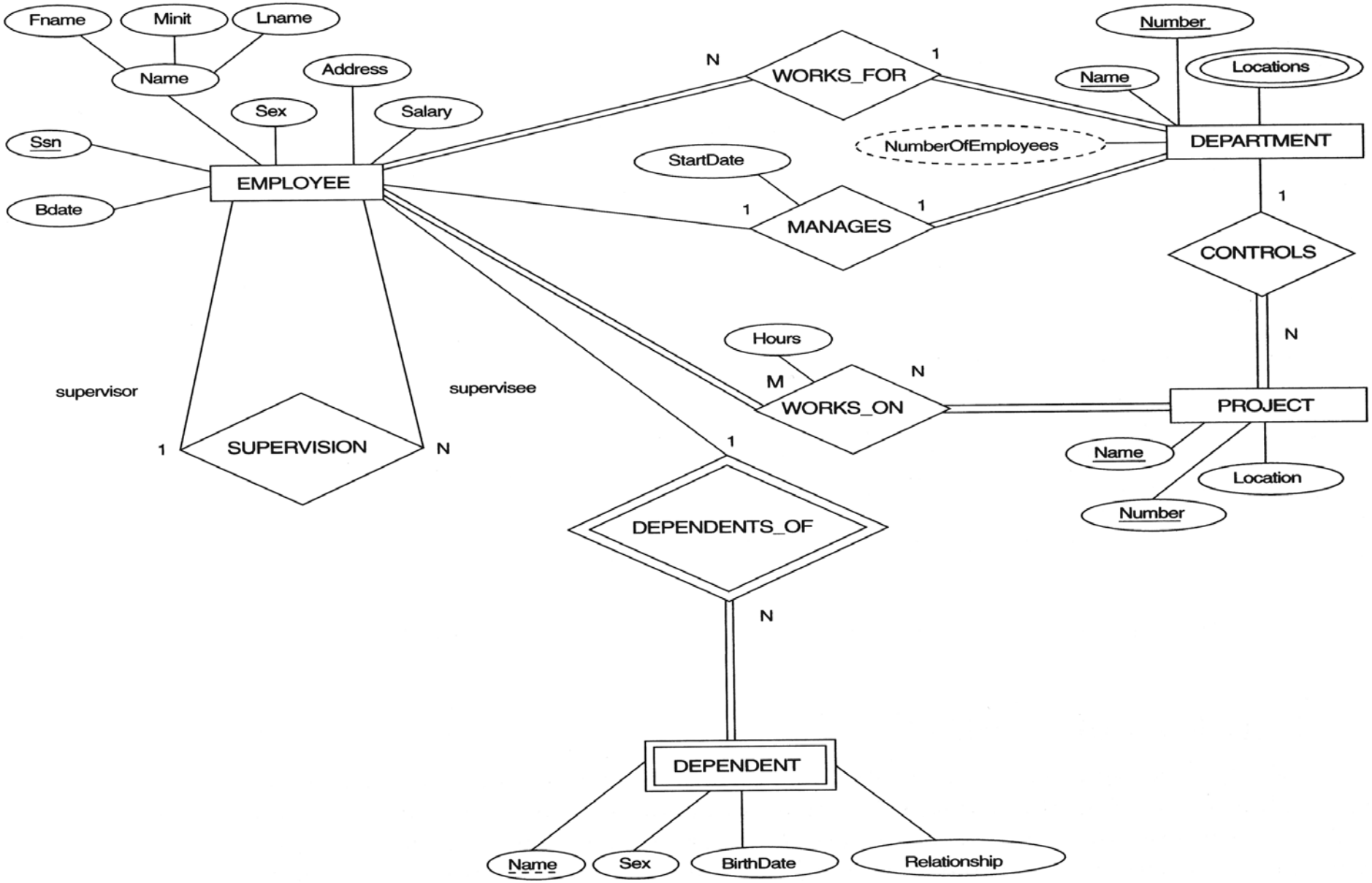
- Types of Attributes
  - In general, composite and multi-valued attributes may be nested arbitrarily to any number of levels although this is rare

    →E.g., PreviousDegrees of a STUDENT is a composite multi-valued attribute denoted by

    { PreviousDegrees (College, Year, Degree, Field) }
  - Derived Attribute

    →Attribute that represents a value that is derivable from value of a related attribute, or set of attributes, not necessarily in the same entity type

Example COMPANY Database

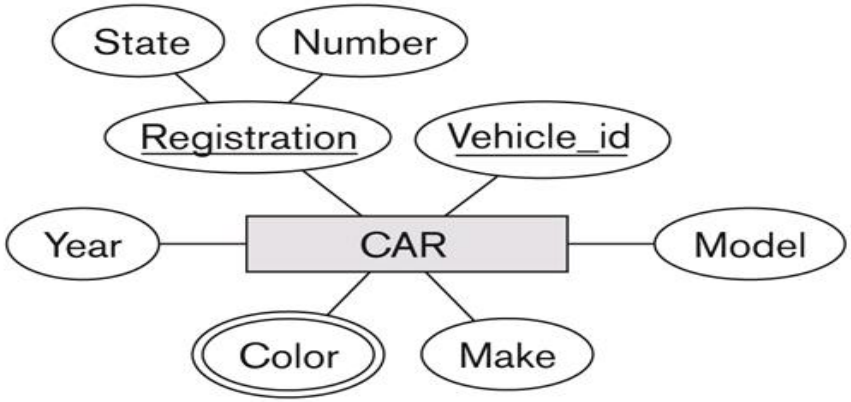


ER Model Concepts

- Entity Types and Key Attributes
  - Entities with the same basic attributes are grouped or typed into an entity type. For example, the EMPLOYEE entity type or the PROJECT entity type
  - 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. For example, VehicleTagNumber is a key of the CAR entity type with components (Number, State)
  - An entity type may have more than one key. For example, the CAR entity type may have two keys:
    →VehicleIdentificationNumber (popularly called VIN) and
    →VehicleTagNumber (Number, State), also known as license_plate number

Entity Type CAR with two keys and a corresponding Entity Set

(a)



(b)

CAR

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

CAR_1

((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 2004 {red, black})

CAR_2

((ABC 123, NEW YORK), WP9872, Nissan Maxima, 4-door, 2005, {blue})

CAR_3

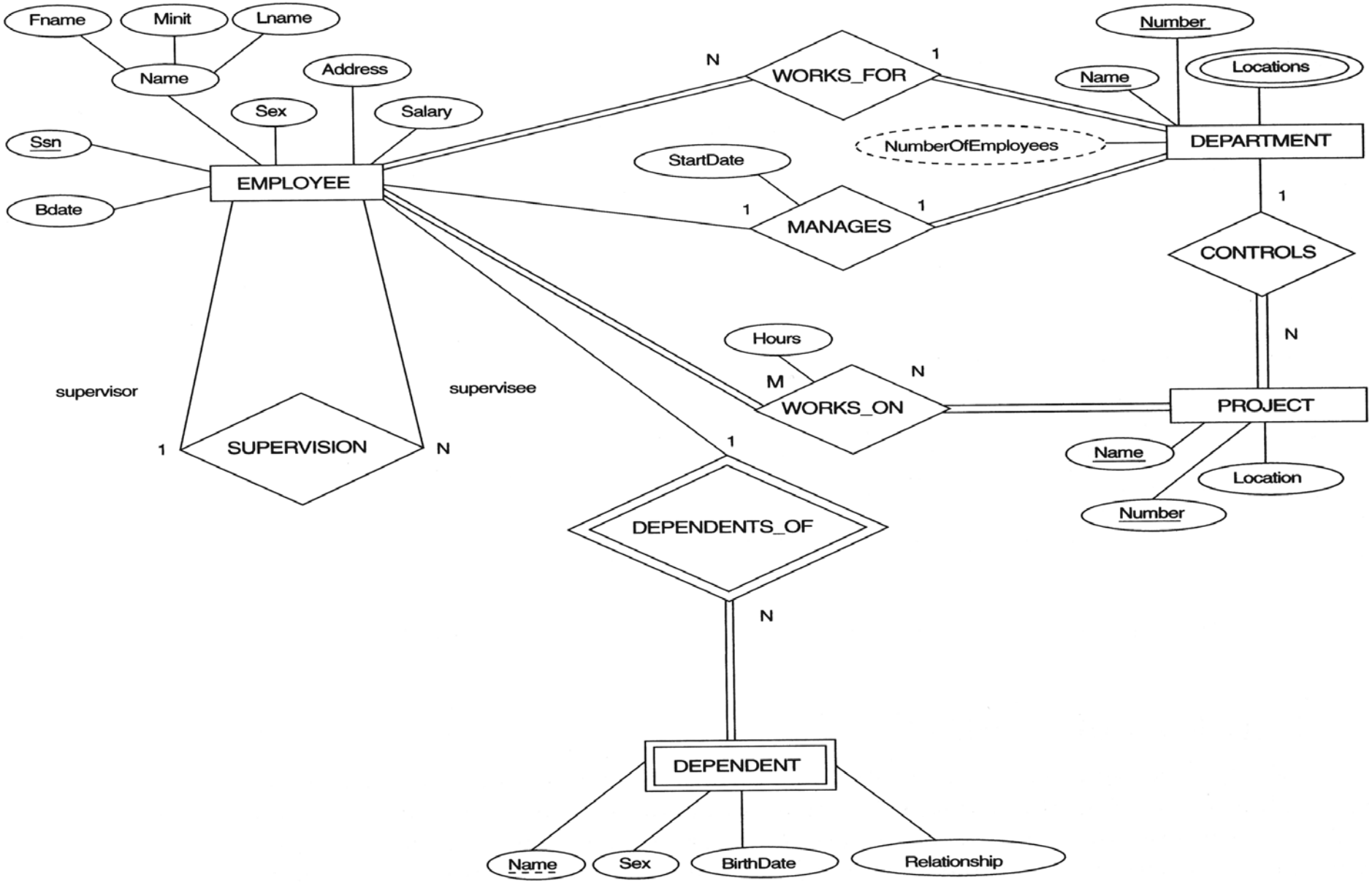
((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4-door, 2002, {white, blue})

•
•
•

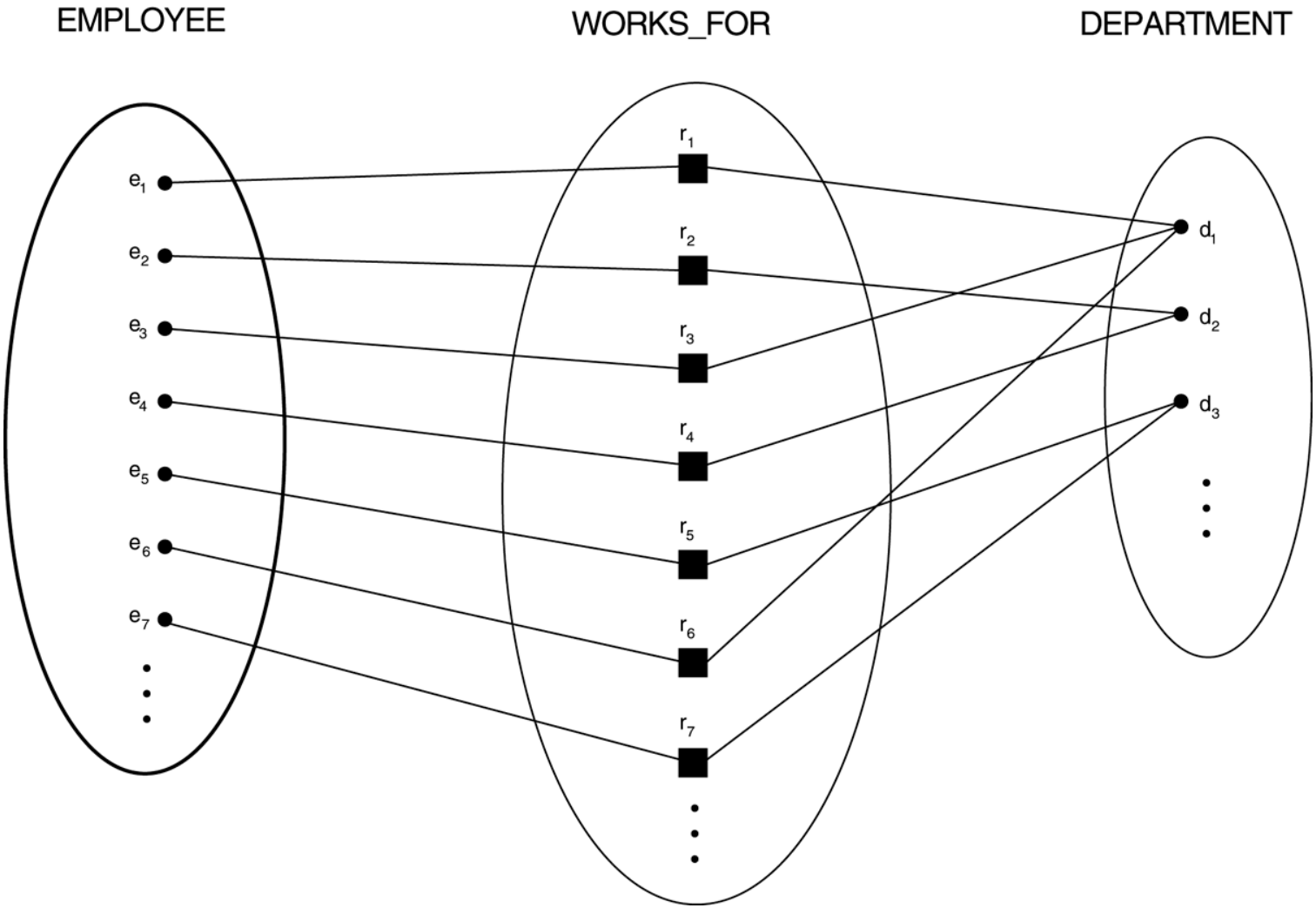
ER Model Concepts

- 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 or EMPLOYEE Franklin Wong manages the Research DEPARTMENT
  - Relationships of the same type are grouped or typed into a relationship type. For example, the WORKS_FOR relationship type in which EMPLOYEEs & DEPARTMENTs participate, or the MANAGES relationship type in which EMPLOYEEs & DEPARTMENTs participate
  - The degree of a relationship type is the number of participating entity types. Both MANAGES and WORKS_ON are binary relationships

Example COMPANY Database



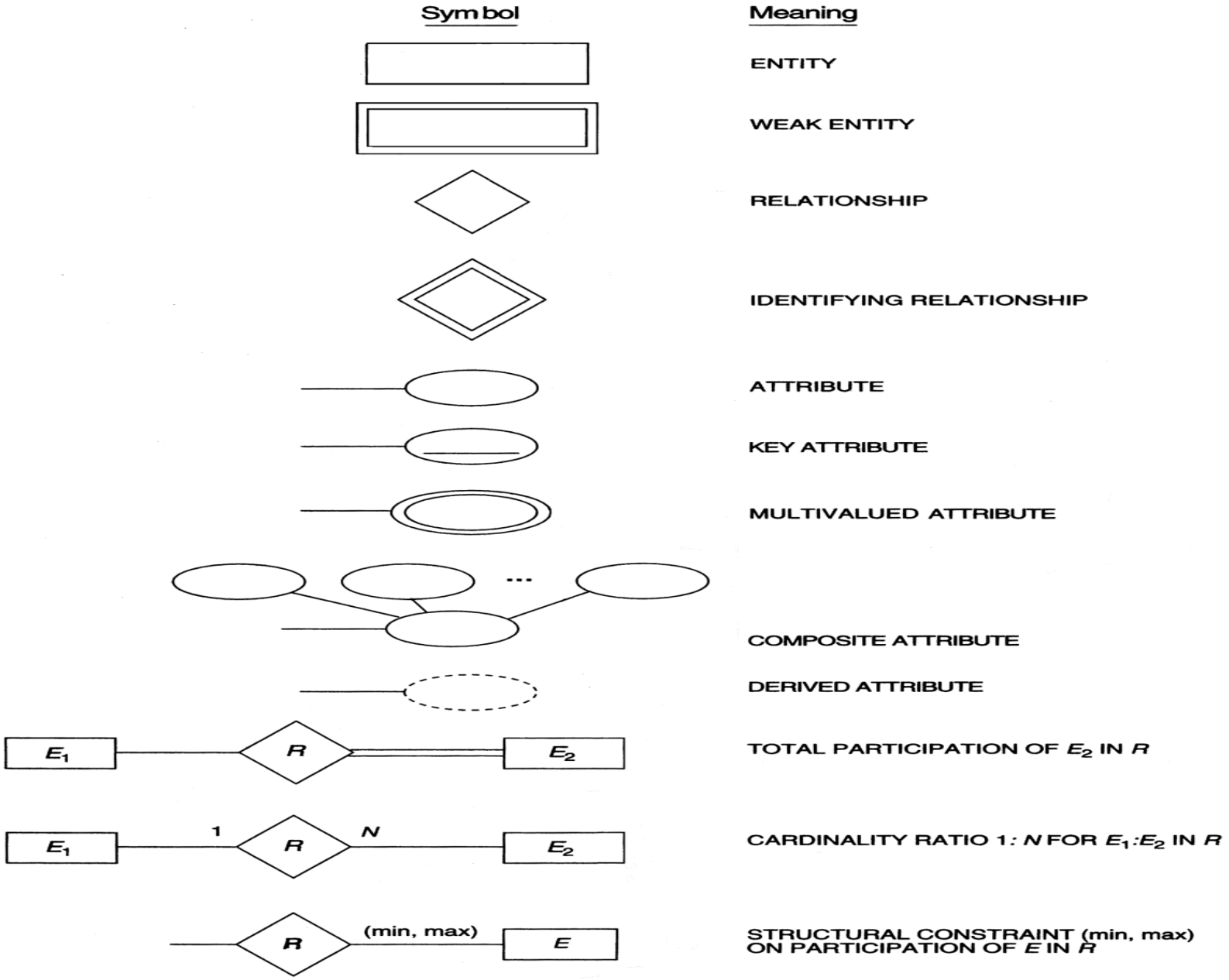
Example relationship instances



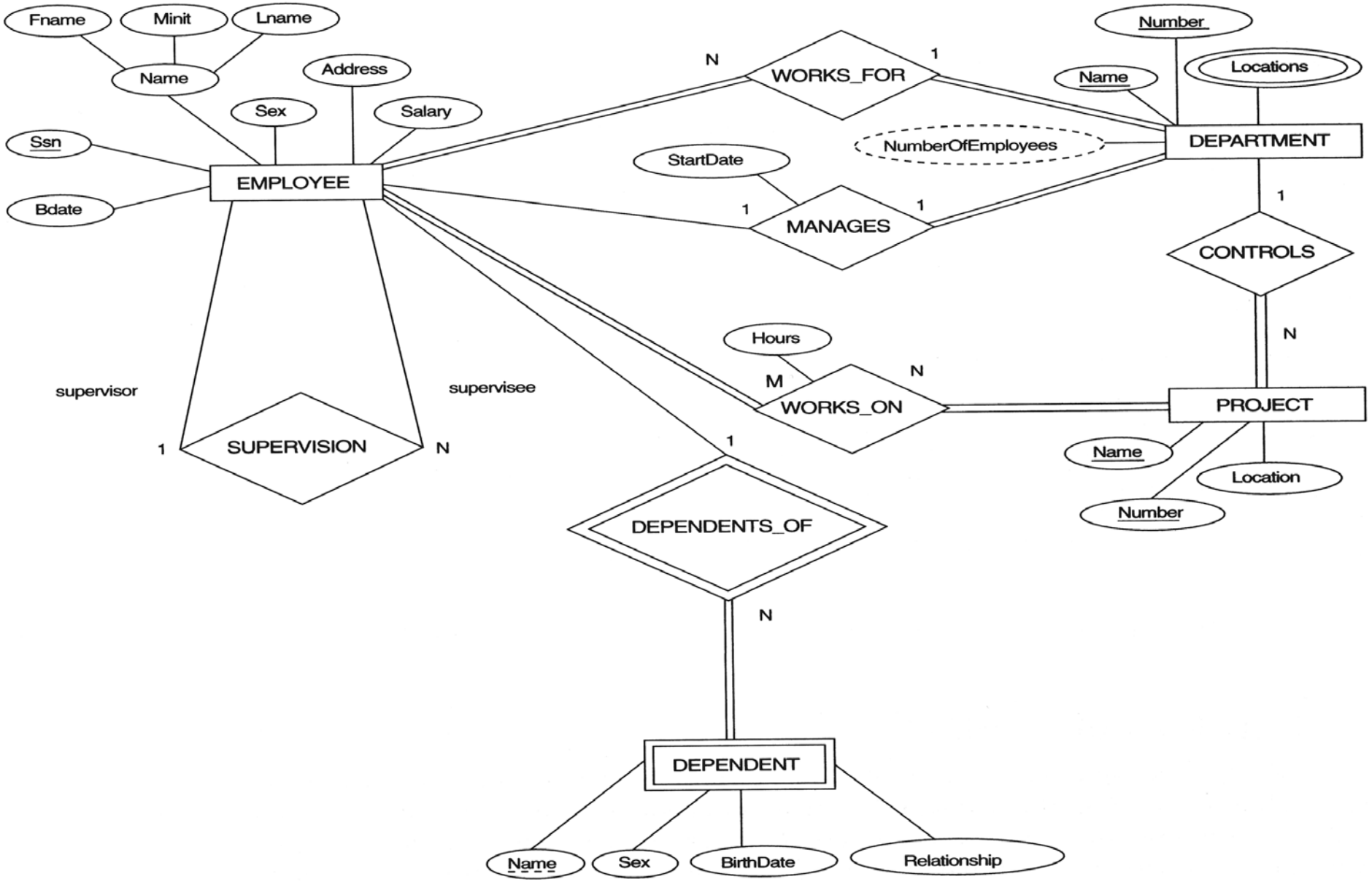
ER Model Concepts

- Relationships and Relationship Types
  - More than one relationship type can exist with the same participating entity types. For example, MANAGES and WORKS_FOR are distinct relationships between EMPLOYEE and DEPARTMENT, but with different meanings and different relationship instances

Summary of the Notation for ER Diagrams



Example COMPANY Database



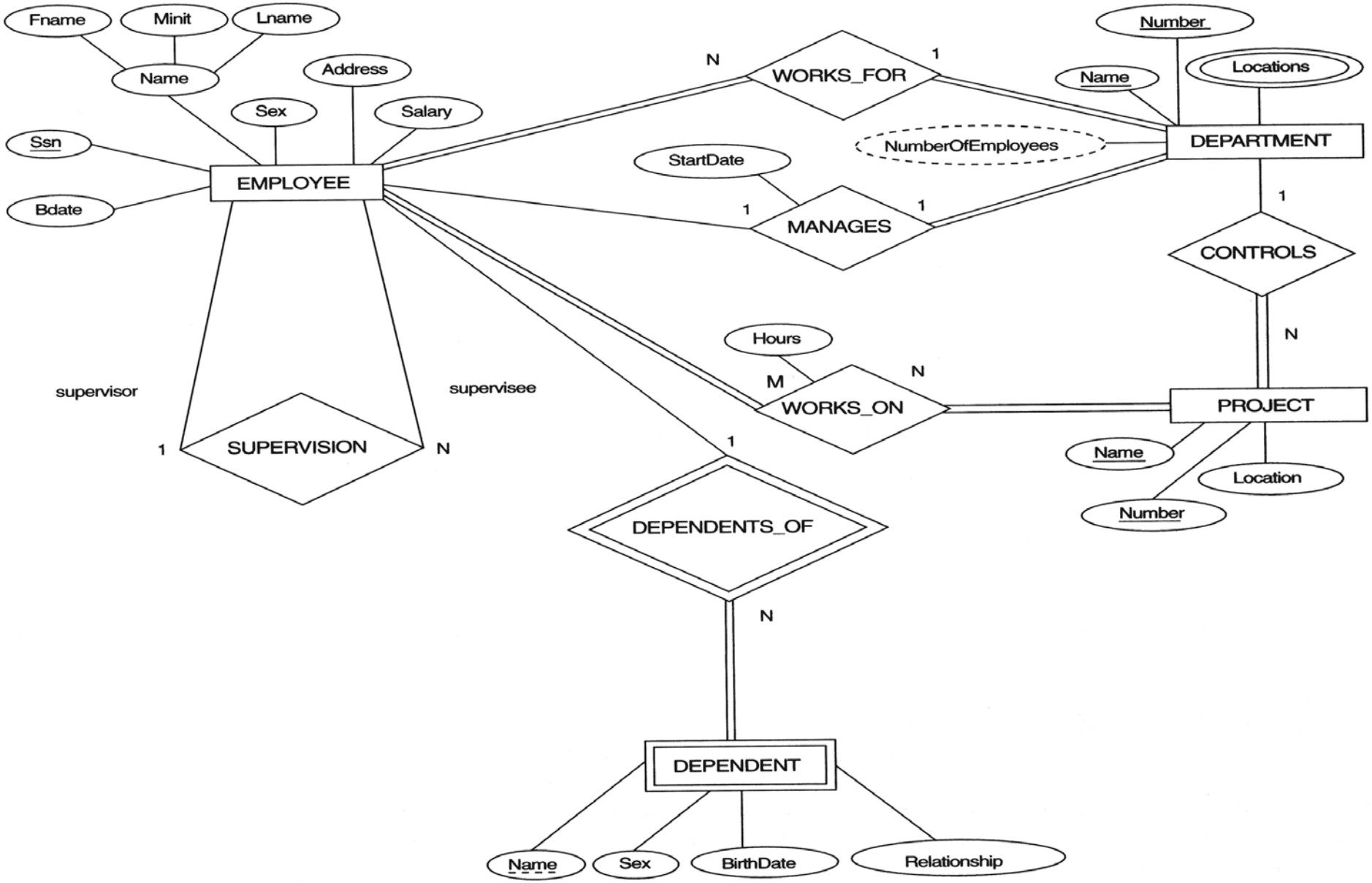
ER Model Concepts

- Attributes of Relationship Types:
  - A relationship type can have attributes; for example, HoursPerWeek of WORKS_ON; its value for each relationship instance describes the number of hours per week that an EMPLOYEE works on a PROJECT

ER Model Concepts

- Weak Entity Types
  - 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: Suppose that a DEPENDENT entity is identified by the dependent's first name (unique wrt. each EMPLOYEE), *and* the specific EMPLOYEE that the dependent is related to. DEPENDENT is a weak entity type with EMPLOYEE as its identifying entity type via the identifying relationship type DEPENDENT_OF

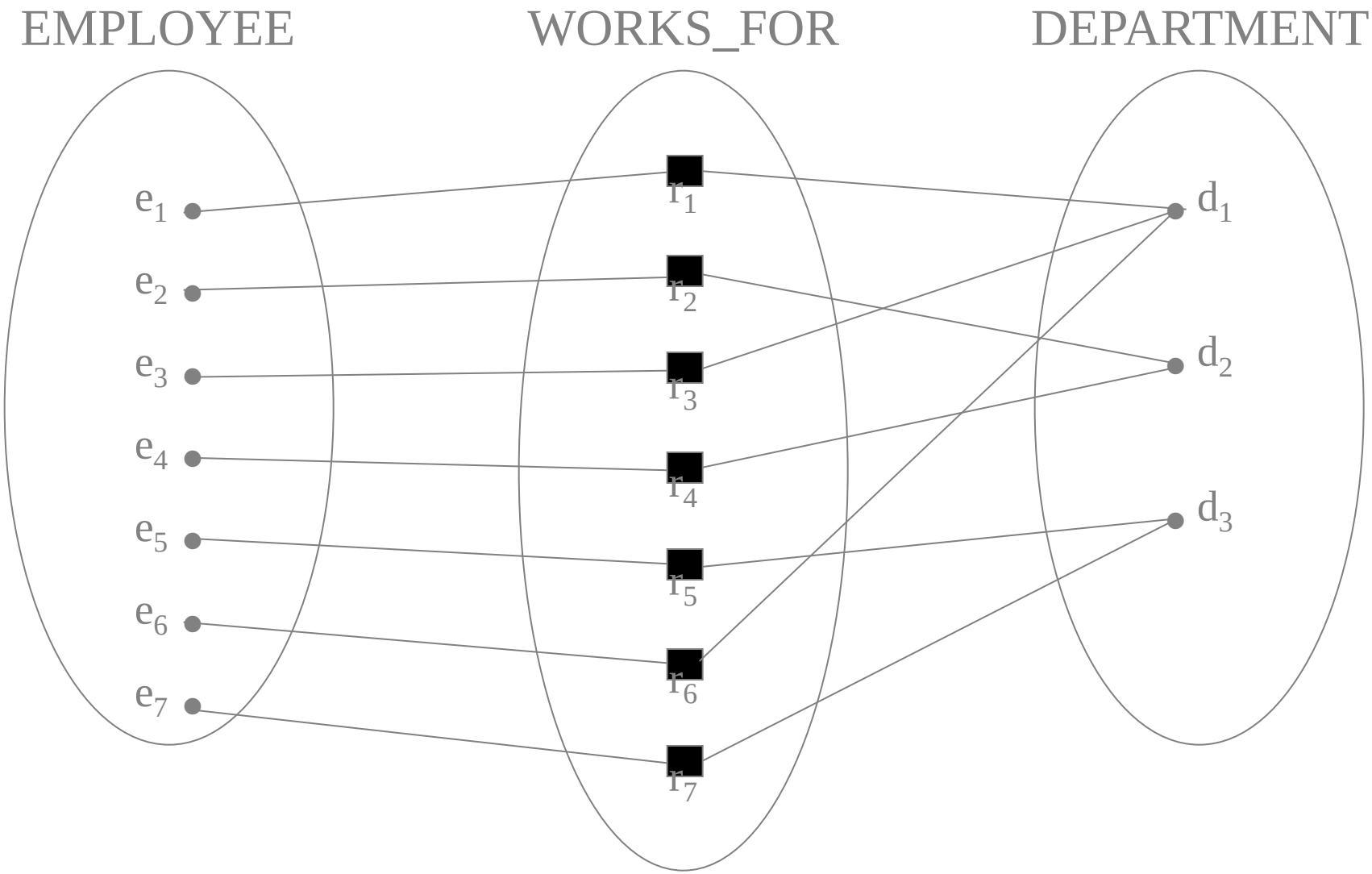
Example COMPANY Database



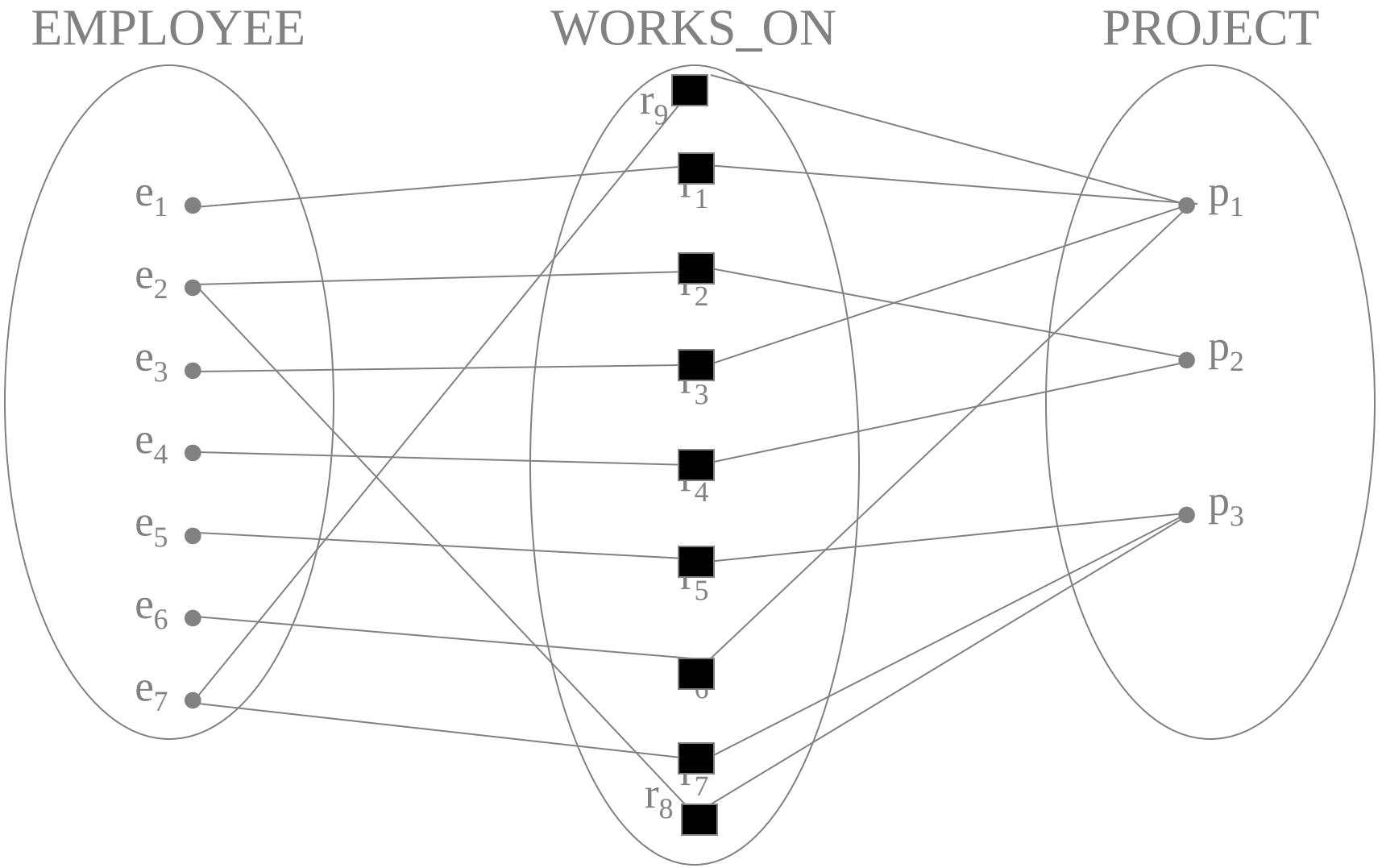
ER Model Concepts

- **Structural constraints:** one way to express semantics of relationship: cardinality ratio and membership class
- **Cardinality ratio (functionality)**: It specifies the number of relationship instances that an entity can participate in a binary relationship
  - one-to-one (1:1)
  - one-to-many (1:M) or many-to-one (M:1)
  - many-to-many (M:N)
- An example of a 1:1 binary relationship is MANAGES which relates a department entity to the employee who manages that department. This represents the miniworld constraints that an employee can manage only one department and that a department has only one manager
- Relationship types of degree 2 are called binary.Relationship types of degree 3 are called **ternary** and of degree n are called **n-ary.** In general, an n-ary relationship *is not* equivalent to n binary relationships (reading suggestion !!)

One-to-many (1:N) or Many-to-one (N:1) RELATIONSHIP



Many-to-many (M:N) RELATIONSHIP



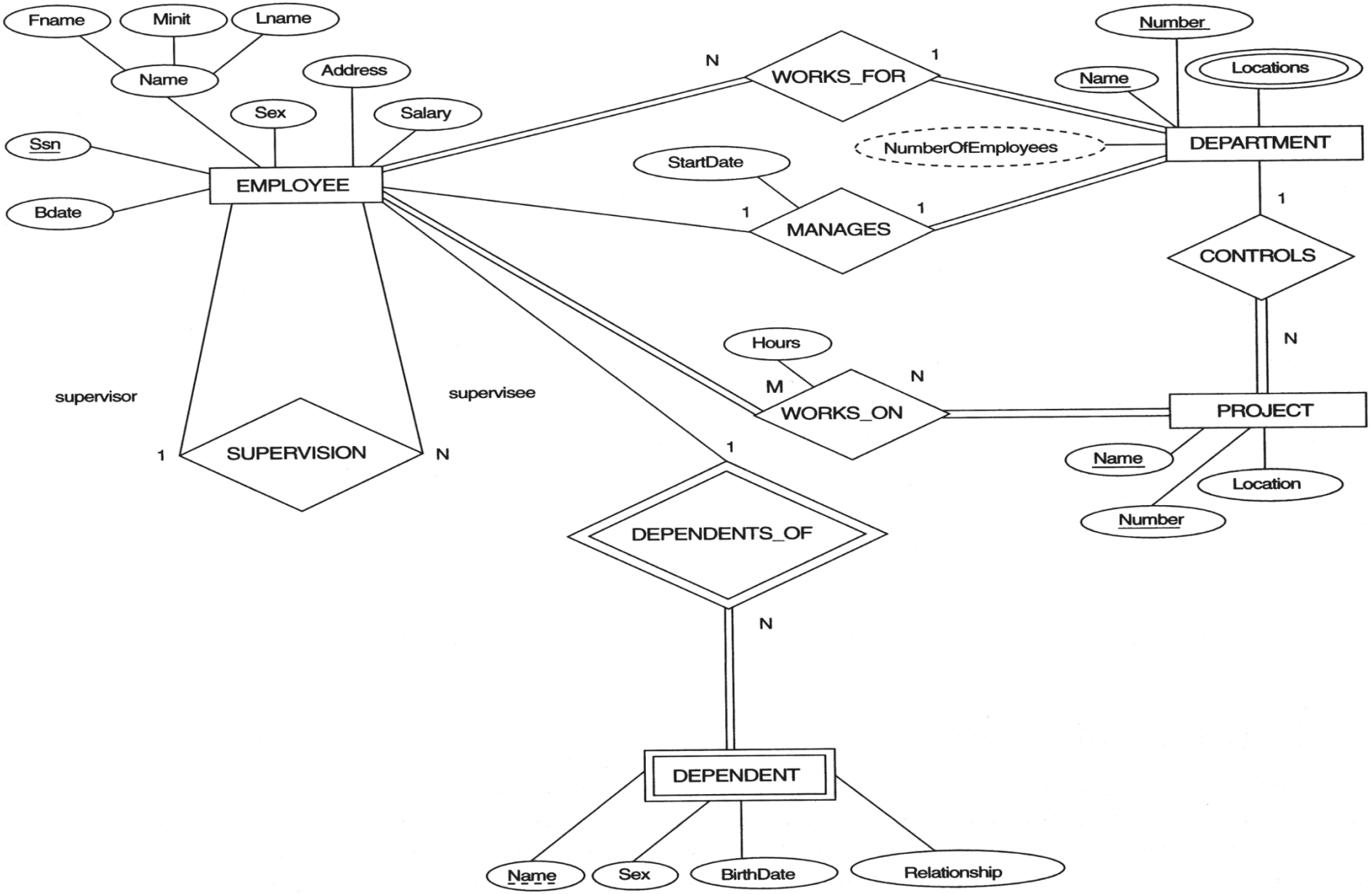
ER Model Concepts

- Membership class (participation constraint):
  - **Mandatory (total participation)** - every instance of a participating entity type must participate in the relationship. Example: ATTEND relationship between STUDENTS and COURSE
  - **Optional (partial participation)** - not every instance of a participating entity type must participate in the relationship. Example: OFFER relationship between SCHOOL and MODULE is optional for SCHOOL but mandatory for MODULE
- Notation:
  - **Cardinality ratio** (of a binary relationship): 1:1, 1:N, N:1, or M:N

    SHOWN BY PLACING APPROPRIATE NUMBER ON THE LINK
  - **Participation constraint** (on each participating entity type): total (called existence dependency) or partial.

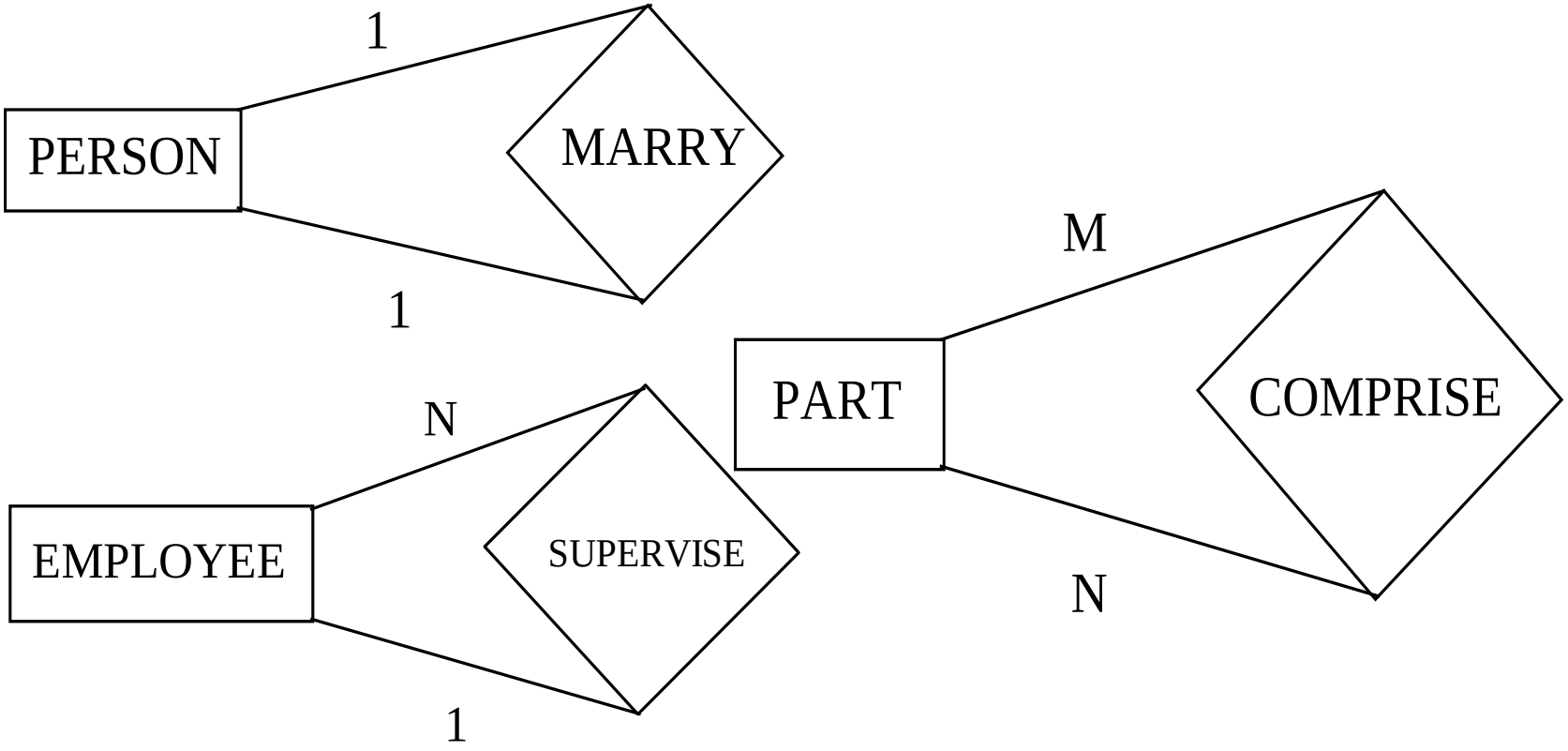
    IN ER DIAGRAMS, TOTAL PARTICIPATION IS DISPLAYED AS A *DOUBLE LINE* CONNECTING THE PARTICIPATING ENTITY TYPE TO THE RELATIONSHIP, WHEREAS PARTIAL PARTICIPATION IS REPRESENTED BY A *SINGLE LINE*

Example COMPANY Database



ER Model Concepts

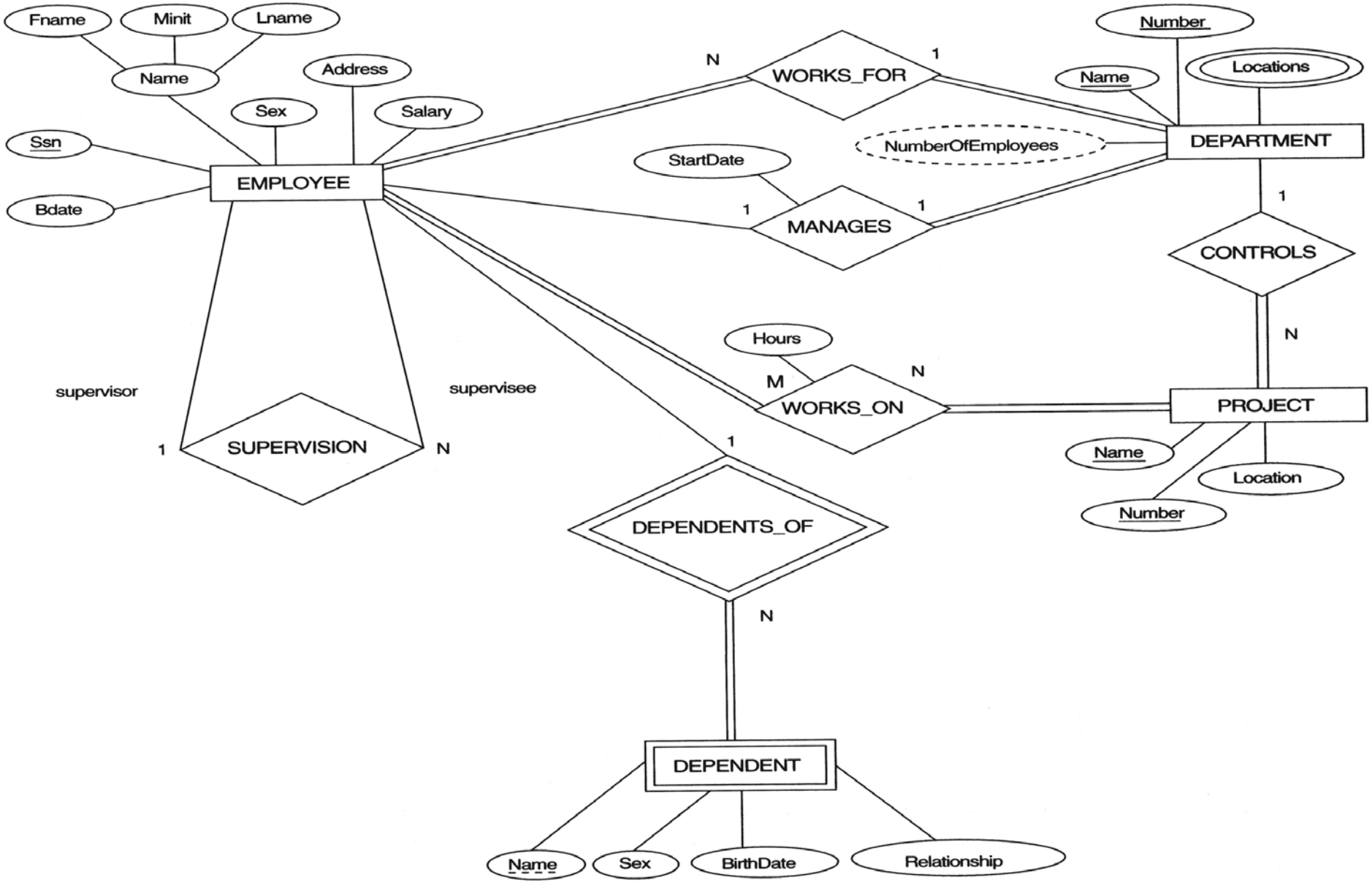
- Recursive relationships (involuted relationship): relationship among different instances of the same entity



ER Model Concepts

- Recursive relationships:
  - Both participations are same entity type in different roles
  - For example, SUPERVISION relationships between EMPLOYEE (in role of supervisor or boss) and (another) EMPLOYEE (in role of subordinate or worker)
  - In following figure, first role participation labeled with 1 and second role participation labeled with 2
  - In ER diagram, need to display role names to distinguish participations

Example COMPANY Database



Outline

- What is ER Model? And Why?
- Overview of Database Design Process
- Example COMPANY Database
- ER Model Concepts
- ER Diagram
- Alternative Diagrammatic Notations
- Problems with ER Models
- Reading Suggestion:
  - [1]: Chapter 3
  - A. Badia: ”Entity-Relationship Modeling Revisited”, SIGMOD Record, 33(1), March 2004, 77-82

ER Diagram

- An ER model can be expressed in the form of the ER diagram
  - An entity type is represented by a rectangular box
  - A relationship is represented by a diamond-shaped box
  - Relationships are linked to their constituent entity types by arcs
  - The functionality of a relationship is indicated on the arc
  - Attributes of entity types/relationships, and membership classes of entity types are listed separately from the diagram
  - The key attribute(s) is <u>underlined</u>

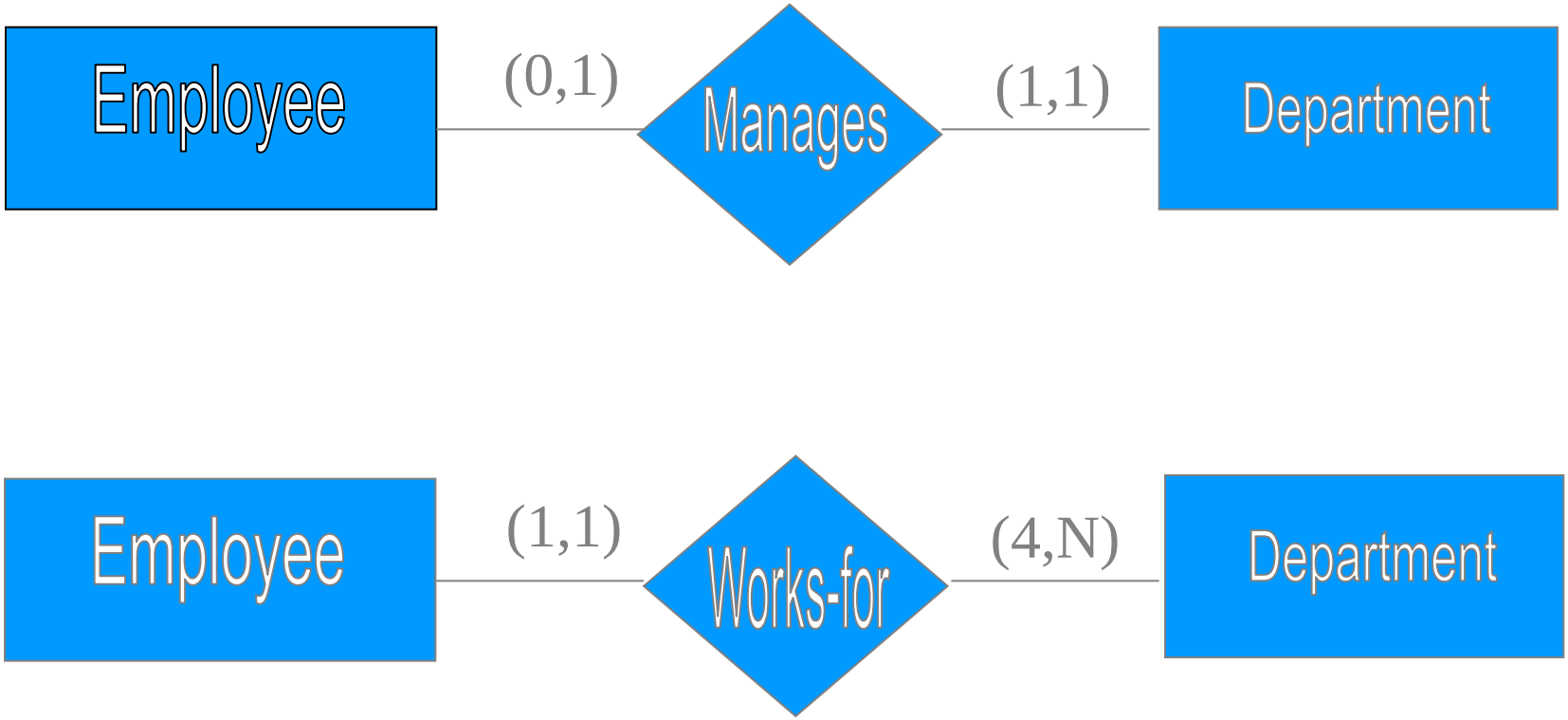
ER Diagram

(min, max) notation for relationship structural constraints

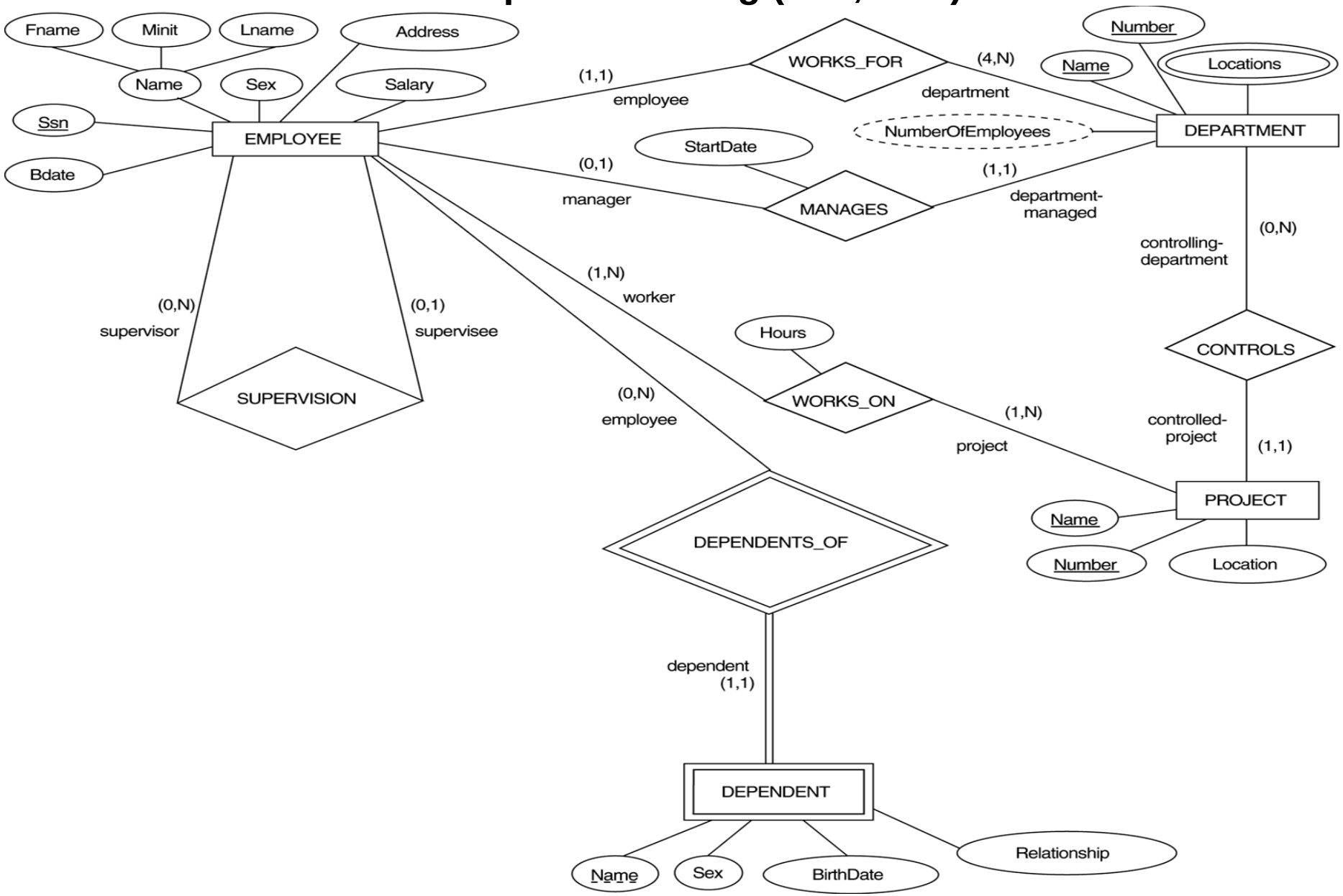
- Specified on *each participation* of an entity type E in a relationship type R
- Specifies that each entity e in E participates in *at least* min and *at most* max relationship instances in R
- Default(no constraint): min=0, max=n
- Must have min≤max, min≥0, max ≥1
- Derived from the knowledge of mini-world constraints
- **Examples:**
  - A department has *exactly one* manager and an employee can manage *at most one* department
    - →Specify (0,1) for participation of EMPLOYEE in MANAGES
    - →Specify (1,1) for participation of DEPARTMENT in MANAGES
  - An employee can work for *exactly one* department but a department must have *at least 4 employees*
    - →Specify (1,1) for participation of EMPLOYEE in WORKS_FOR
    - →Specify (4,n) for participation of DEPARTMENT in WORKS_FOR

ER Diagram

(min, max) notation for relationship structural constraints



ER diagrams for the COMPANY schema, with structural constraints specified using (min, max) notation

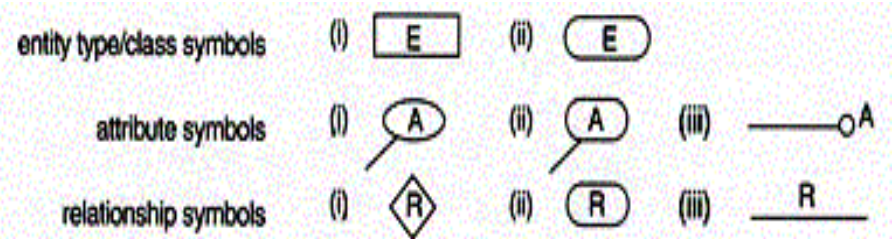


Alternative Diagrammatic Notations

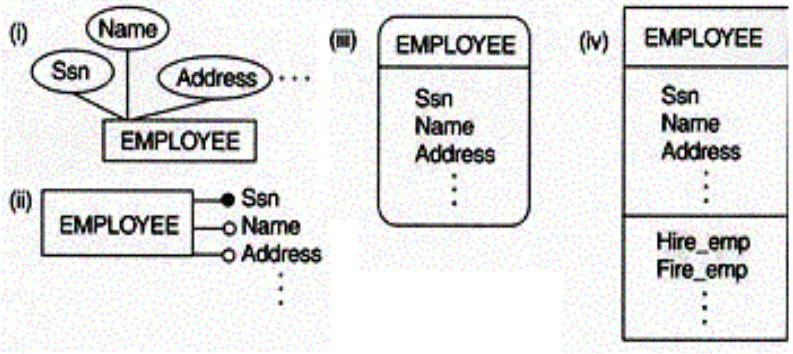
- Current use (in this class):
  - Chen notation
- Some others:
  - Crow Feet notation
  - UML (Unified Modeling Language): e.g., Rational Rose

Alternative Diagrammatic Notations

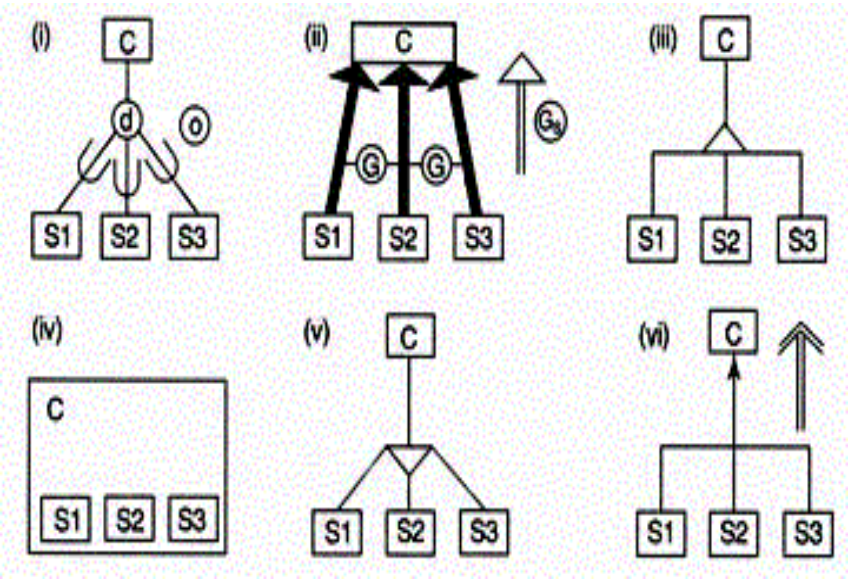
Symbols for entity type / class, attribute and relationship



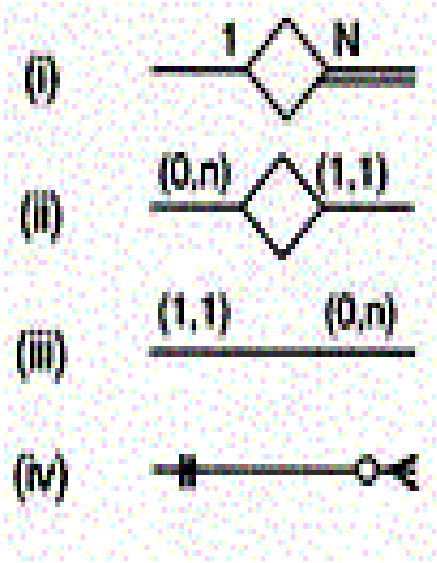
Displaying attributes



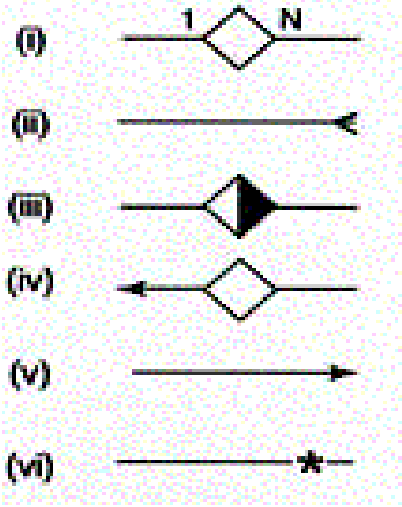
Notations for displaying specialization / generalization



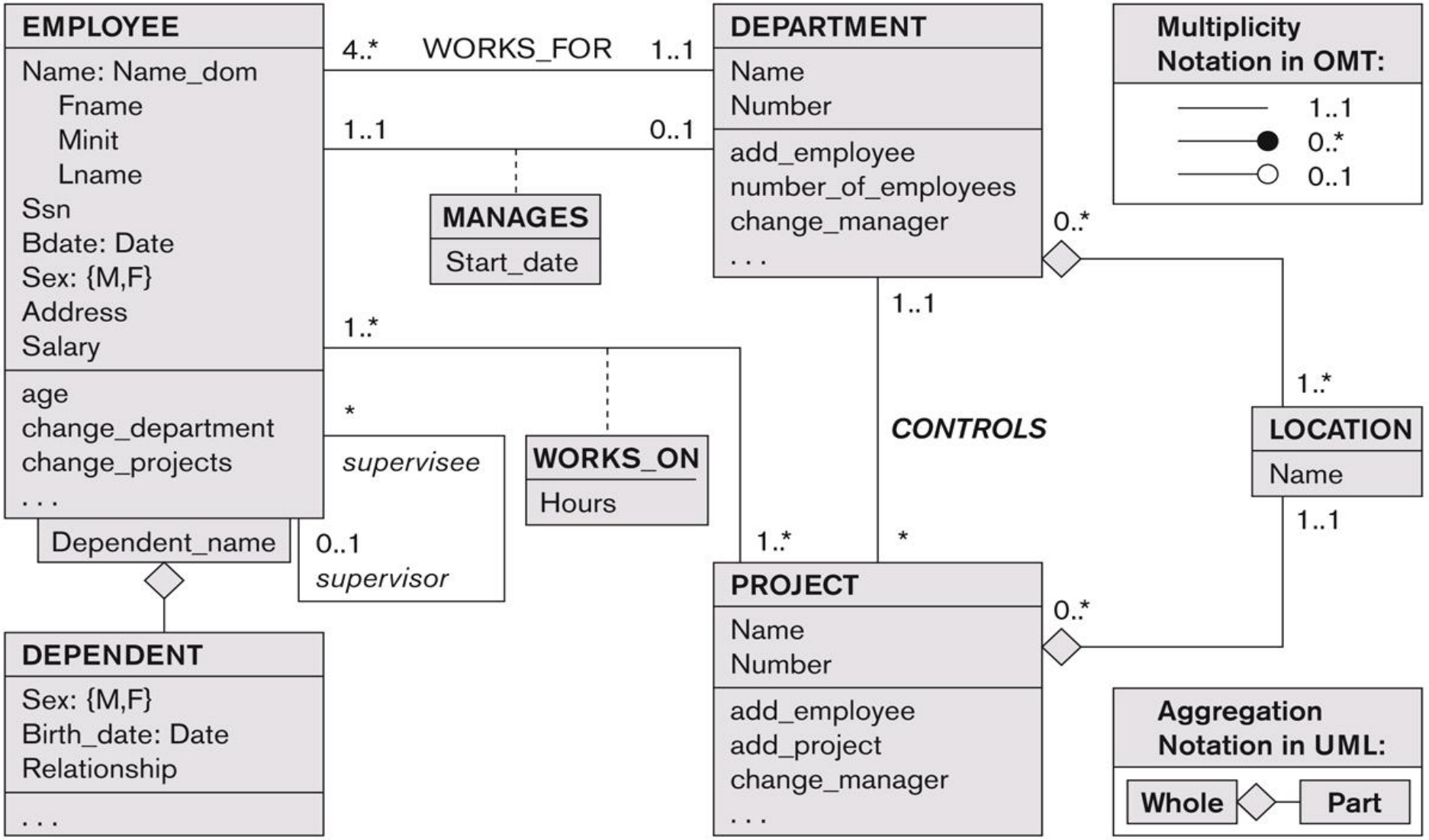
Various (min, max) notations



Displaying cardinality ratios



The COMPANY conceptual schema in UML class diagram notation.



Outline

- What is ER Model? And Why?
- Overview of Database Design Process
- Example COMPANY Database
- ER Model Concepts
- ER Diagram
- Alternative Diagrammatic Notations
- Problems with ER Models
- Reading Suggestion:
  - [1]: Chapter 3
  - A. Badia: ”Entity-Relationship Modeling Revisited”, SIGMOD Record, 33(1), March 2004, 77-82

Problems with ER Models

- Problems may arise when designing a conceptual data model called *connection traps*
- Often due to a misinterpretation of the meaning of certain relationships
- Two main types of connection traps are called *fan traps* and *chasm traps*
- Next lecture: you want to talk ? (some initial materials are provided in my HP)

Summary

- What is ER Model? And Why?
- Overview of Database Design Process
- Example COMPANY Database
- ER Model Concepts
- ER Diagram
- Alternative Diagrammatic Notations (UML)
- Problems with ER Models
- Next lecture:
  - Students' presentation (for bonus): send me an email
  - Introduction to E-ERD, and exercises

Q&A