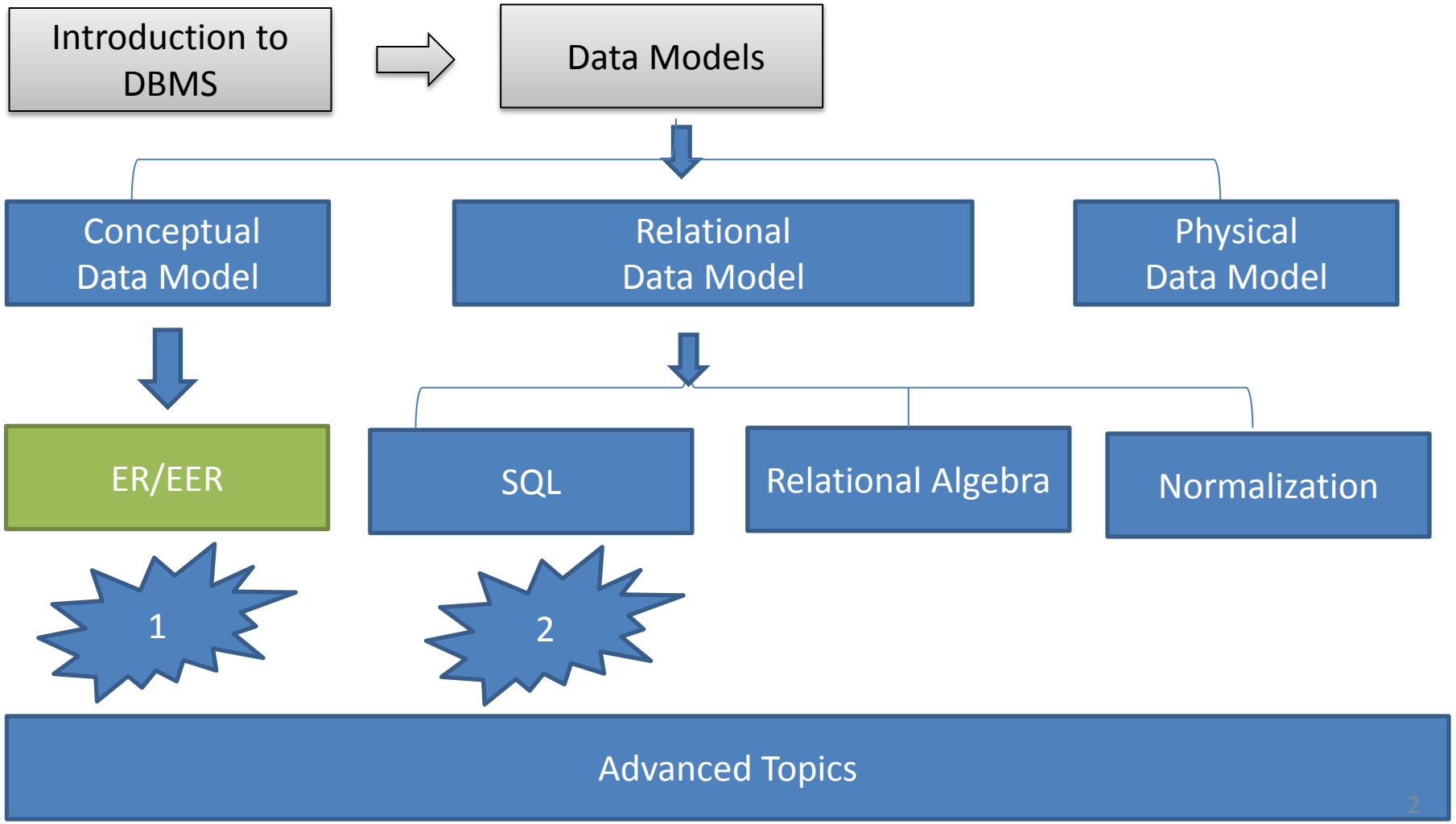
An Introduction to the Database Management Systems

By
Hossein Rahmani

Slides originally by Book(s) Resources

Road Map

(Might change!)



Data Modeling using the Entity Relationship (ER) Model

- High-Level Conceptual Data Models
- Entity Relationship Model and its elements
- From text to ER-schema

Last Session: Three levels of data models

- Conceptual (high-level)
  - for end users / analysts

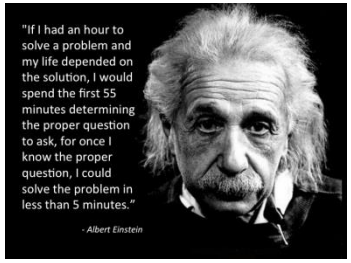
    versus
- Physical (low-level)
  - for programmers
- In between: Representational (implementation) data model
- Today: conceptual model:

  Entity-relationship (ER) model

Entity-Relationship Model

- Conceptual data model
- ER-model is used to create a *conceptual (database) schema*
- Independent of the *representational* model (and of the kind of DBMS)
- The schema will be translated into a *logical schema* (e.g., relational schema, MS-access tables)

Using High-Level Conceptual Data Models for Database Design



- **Requirements collection and analysis**
  - Database designers interview prospective database users to understand and document data requirements
  - Result:
    - **Data requirements**
    - **Functional requirements** of the application

Using High-Level Conceptual Data Models (cont'd.)

- **Conceptual schema**
  - Conceptual design
  - Description of data requirements
  - Includes detailed descriptions of the entity types, relationships, and constraints
  - Transformed from high-level data model into implementation data model

A Sample Database Application

- COMPANY
  - Employees, departments, and projects
  - Company is organized into departments
  - Department controls a number of projects
  - Employee: store each employee's name, Social Security number, address, salary, sex (gender), and birth date
  - Keep track of the dependents of each employee

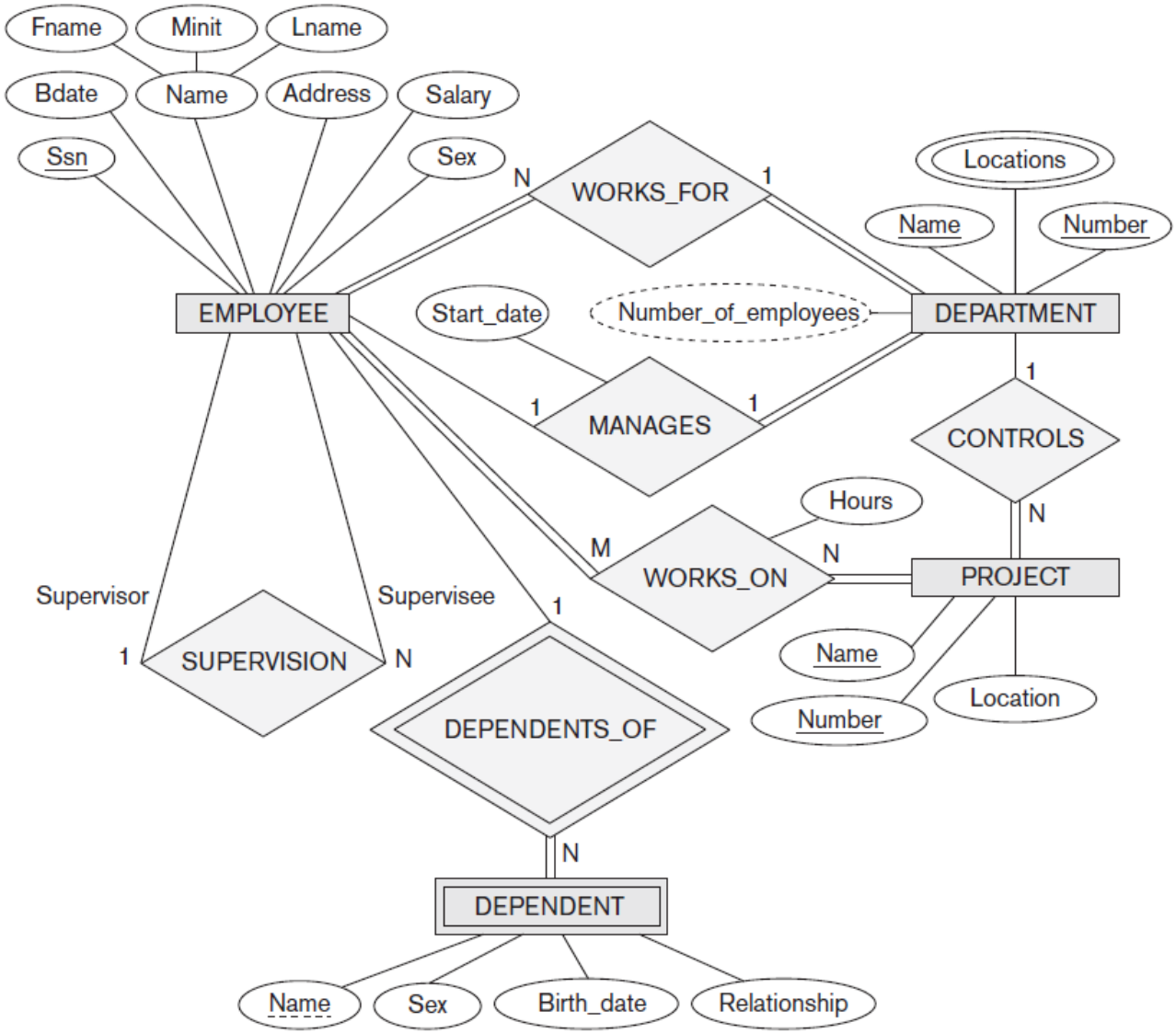
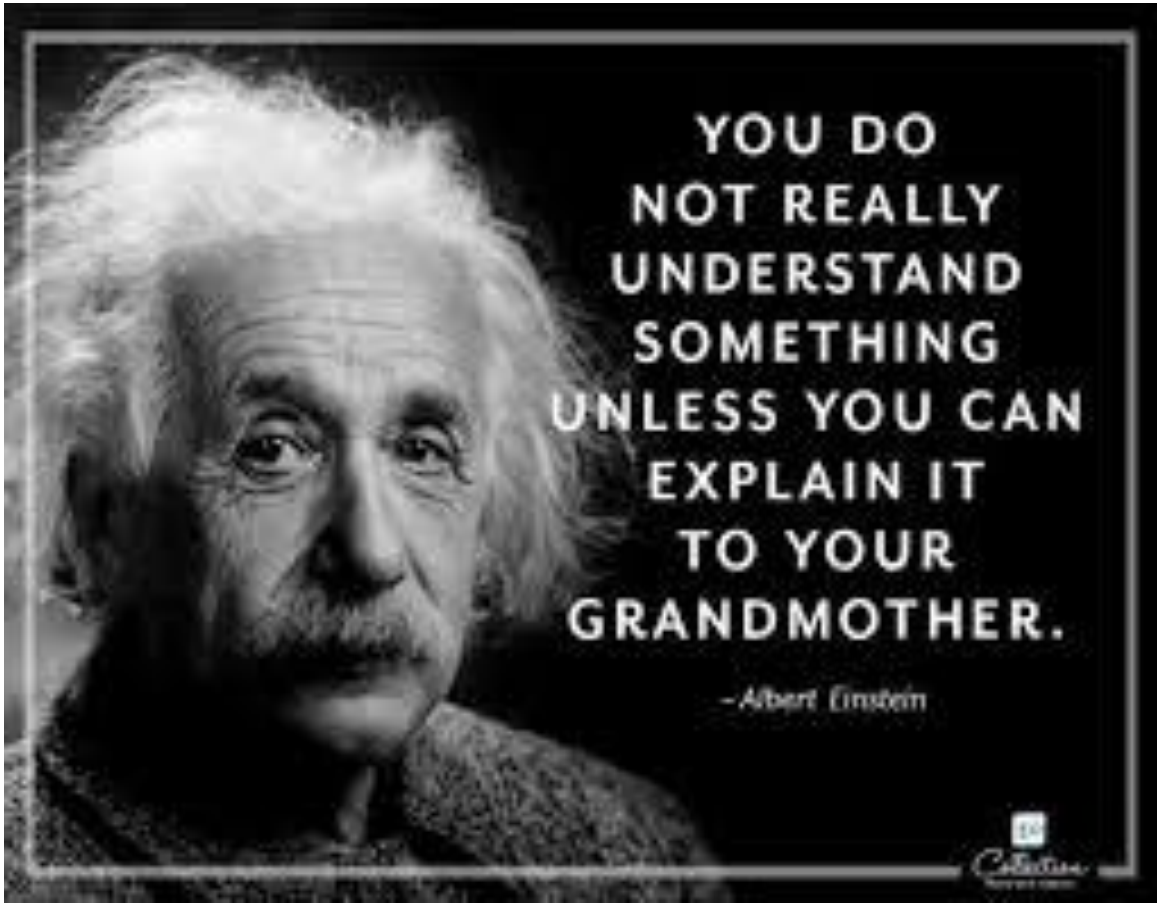


Figure 7.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

What you understand from the previous ER diagram?



Data Modeling using the Entity Relationship (ER) Model

- High-Level Conceptual Data Models
- Entity Relationship Model and its elements
- From text to ER-schema

Entity Relationship Model

- Relevant information from the mini world (universe of discourse) is described in terms of:
  - Entities
  - Attributes
  - Relations

Entities

- An entity in the ER model is the representation of a "thing" in the real world that exists on its own.

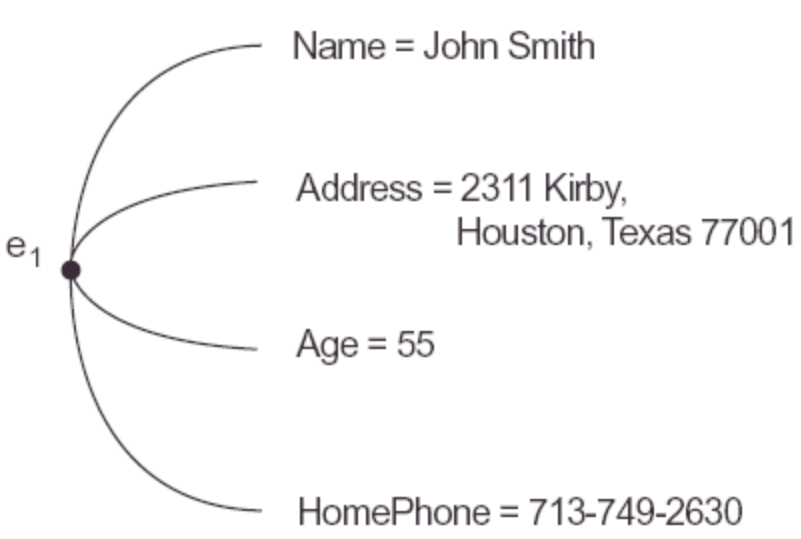
  - Tangibles: a *person, a house, a car*
  - Intangibles : a *company, a job, a course*

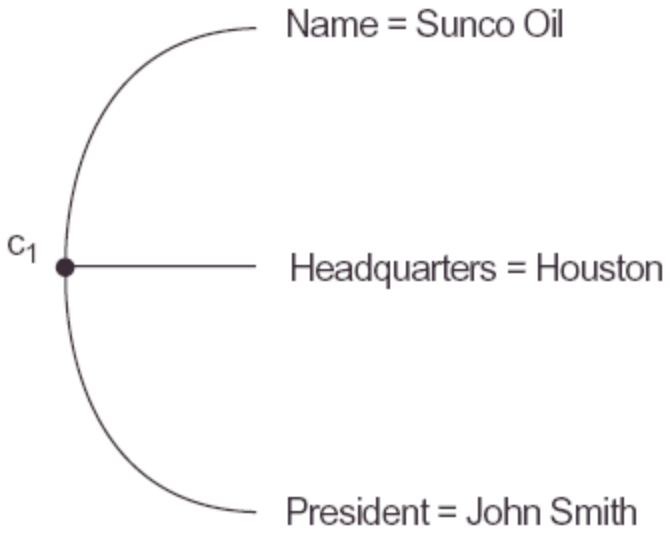
Attributes

- Each entity has *attributes* that describe the intricacies of that specific entity

- A specific entity has *values* for all its attributes

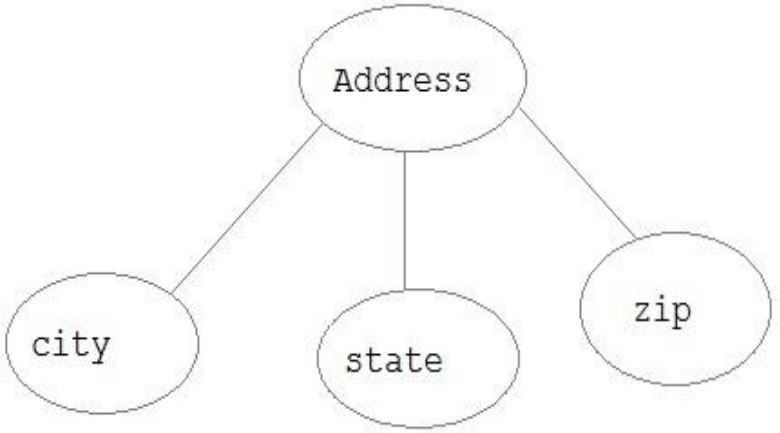
Entities and attributes





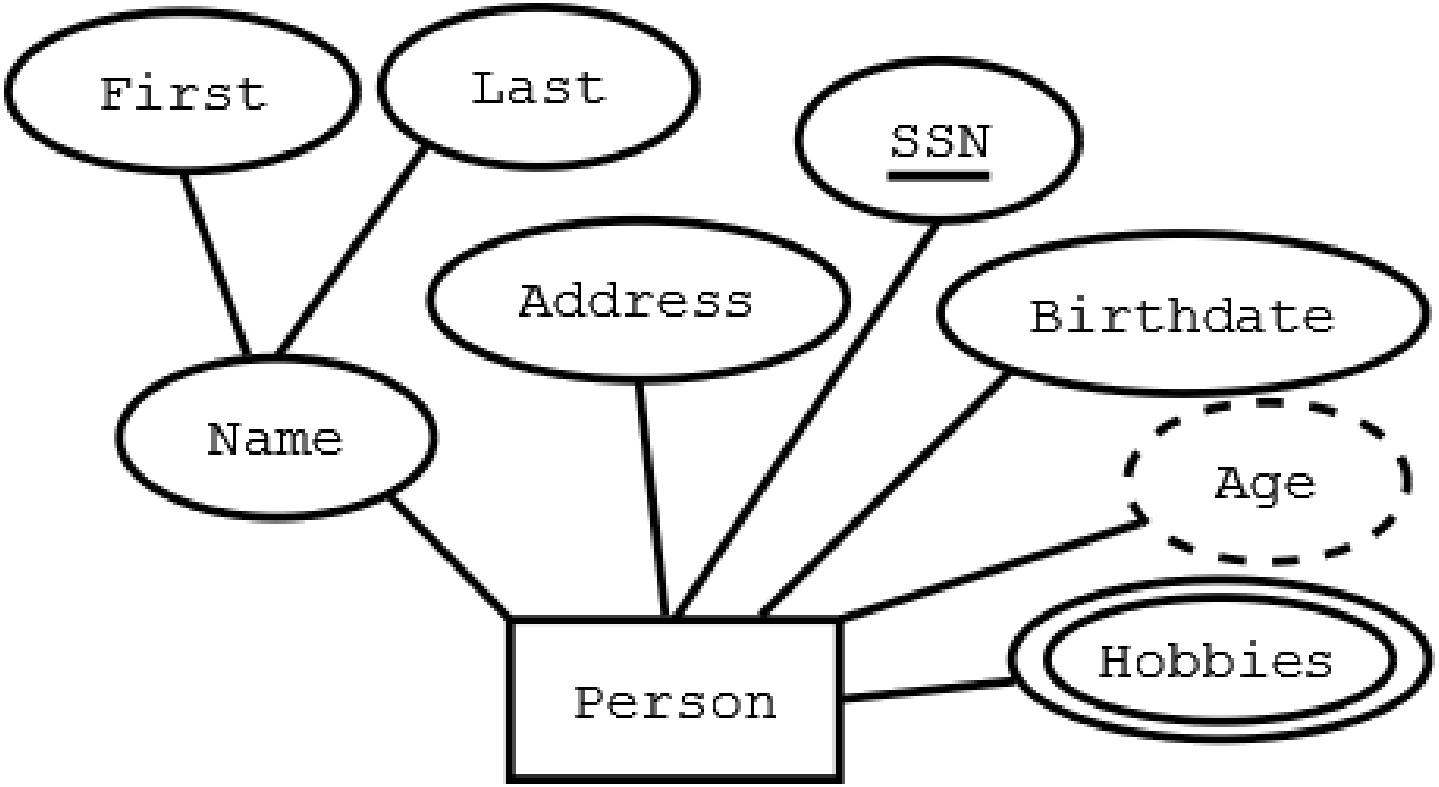
Kinds of attributes

- Composite
  - an attribute that can be split:
    - address = “parkstraat 12a, 5243 GE, Hoogestande”
    - “parkstraat” - “12” - “a” - “5243 GE” - “Hoogestande”
    - attribute-hierarchy
- Atomic
  - cannot be split
    - age = 23



Kinds of attributes

- Single-valued versus multi-valued
  - age = 23 (single-valued)
  - Phone.nr = 043-2235541 and 06-2355534
  - hobby = swimming and piano and party
- Stored versus derived
  - age derived from date of birth
- Complex attributes:
  multi-valued “{}” and composite “()”


First
Last
SSN
Address
Birthdate
Name
Age
Hobbies
Person

Null-values

- An attribute can have a special value *null* (or *nil*). The meaning of the value depends on the context.
- Possible meanings of null:
  - *Not applicable / cannot be determined*
    - *College_degree*
  - *Not relevant*
  - *Applicable but (still) unknown*
  - *Should be known but is missing*

Entity-Types

- In a database there exist groups of *similar* entities:
  - having the same attributes
  - having different values for the attributes

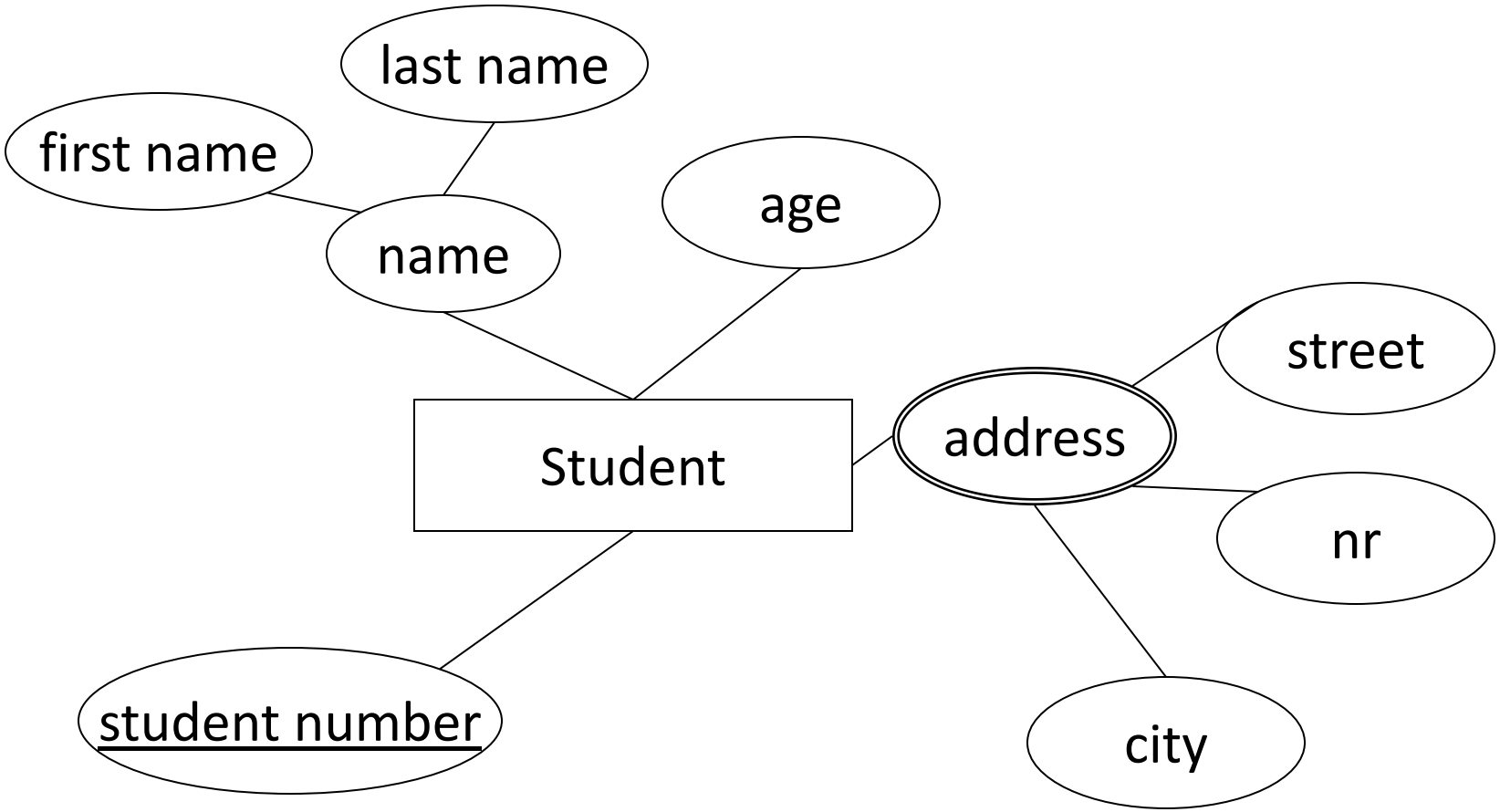
- An **entity-type** represents a *collection of similar entities*.
  (students, houses, companies)

Entity-Type

- In an **ER-schema** entity-types are determined by a unique *name* and a set of *attribute-names*.

- In an **ER-diagram** an entity-type is represented by a rectangle, containing the name, to which ellipses are connected, containing the attribute names. (multivalued: double lines)

Entity-Type


last name
first name
name
age
street
Student
address
nr
student number
city

Key attribute

- A **key** is a collection of attributes of an entity-type for which the *combination of values is unique* for each entity in all possibly occurring entity-sets
  (uniqueness constraint)

- Attributes within a key are called key attributes. (underlined in an ER-diagram)

Key attributes

- An entity-type can have more than one key
- An entity-type can also have no keys (i.e. **weak** entity-type).
- The attributes of any key may never be all null at the same time

Attribute domain

- Every atomic attribute is associated with a value domain
  - natural numbers
  - [30,50]
  - texts of up to 50 characters
  - yes/no
- This is not represented in the ER-diagram (but does belong to the ER-schema!)

Initial Conceptual Design of the COMPANY Database

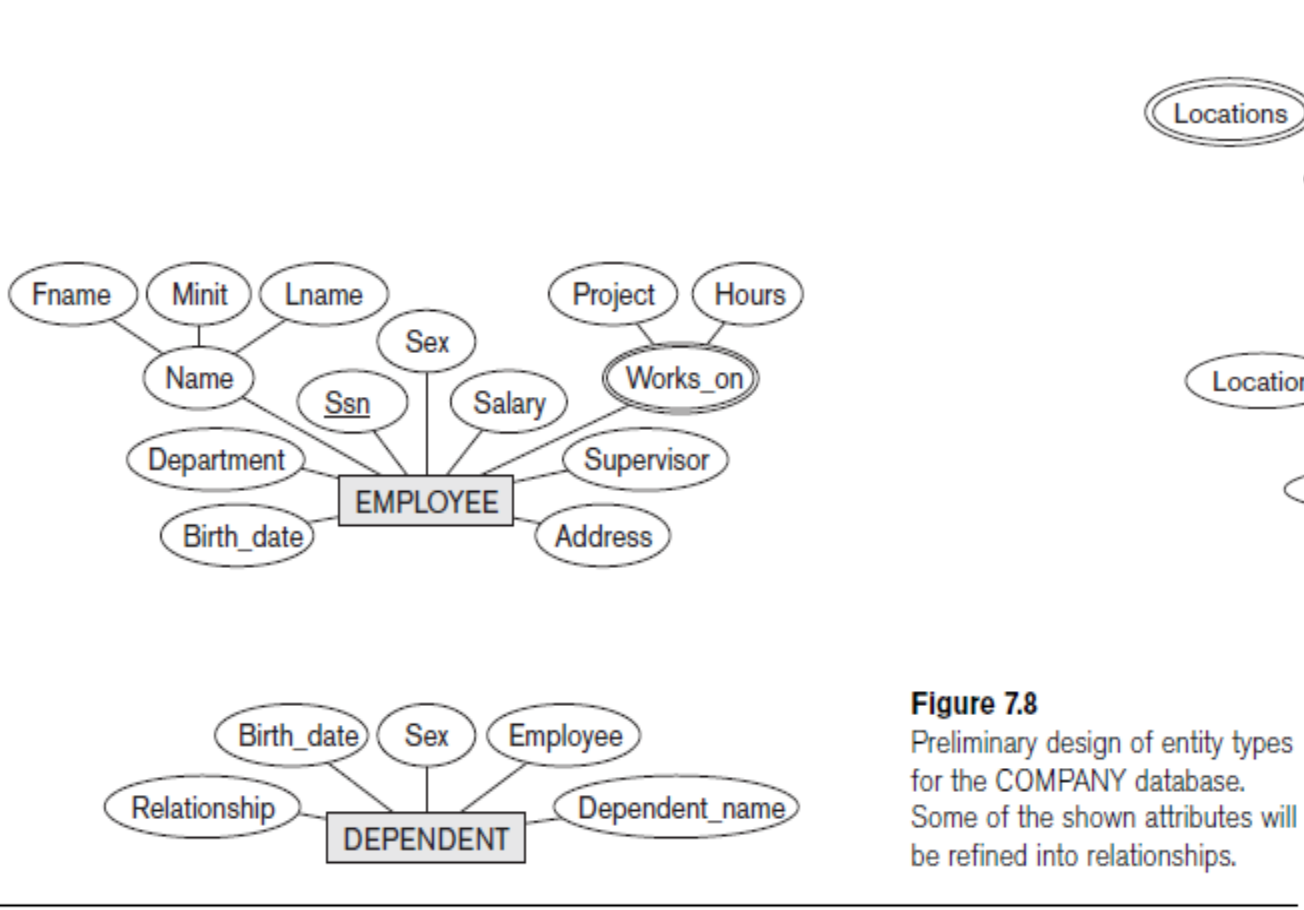


Figure 7.8
Preliminary design of entity types for the COMPANY database. Some of the shown attributes will be refined into relationships.

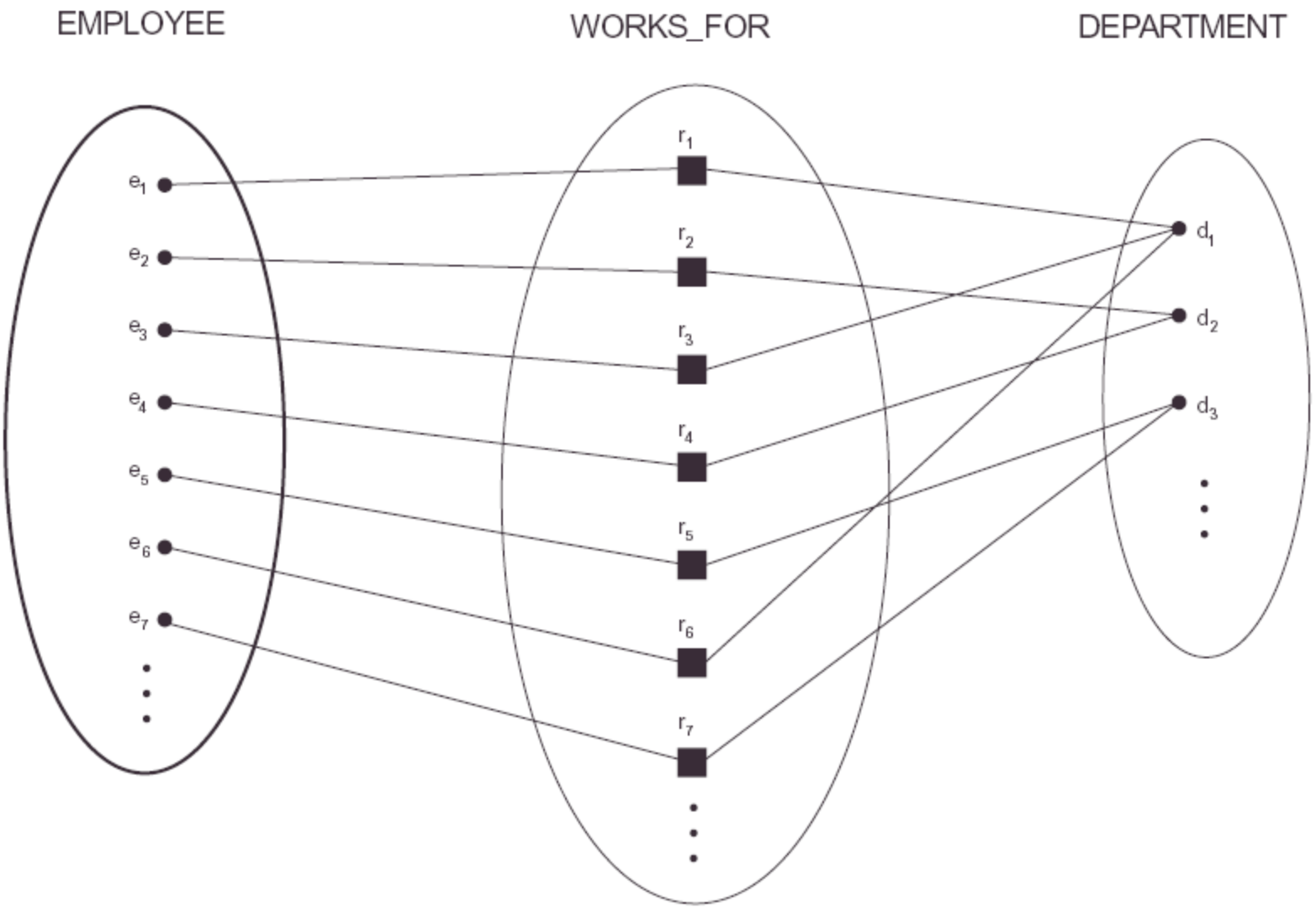
Relations

- A *relation* is the representation of a relevant, existing association between two (or more) entities in the mini world.

  - Student i99883 follows ‘databases’
  - FHS is a faculty of the UM

Relation-Types

- A relation-type between two or more entity-types represents the *set of possible relations* between entities of these entity-types
  - A relation-type *R* on
    participating entity-types $E_1, E_2, .. E_n$
    is a set of
    relation-instances $(e_1, e_2, .. e_n)$
    and is a subset of $E_1 \times E_2 \times .. \times E_n$

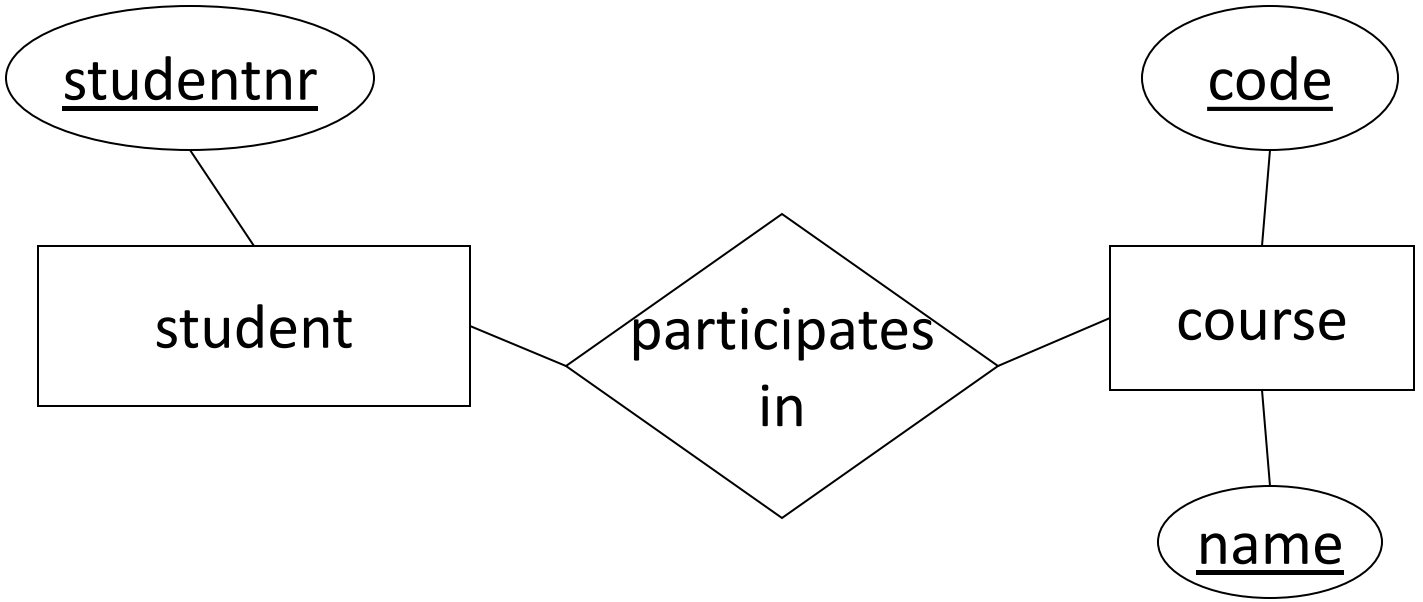

EMPLOYEE
WORKS_FOR
DEPARTMENT
e1
e2
e3
e4
e5
e6
e7
r1
r2
r3
r4
r5
r6
r7
d1
d2
d3

Relation-Types

- Each relation-type has a *name*
- Examples:
  - Faculty is-faculty-of University
  - Student participates-in Course

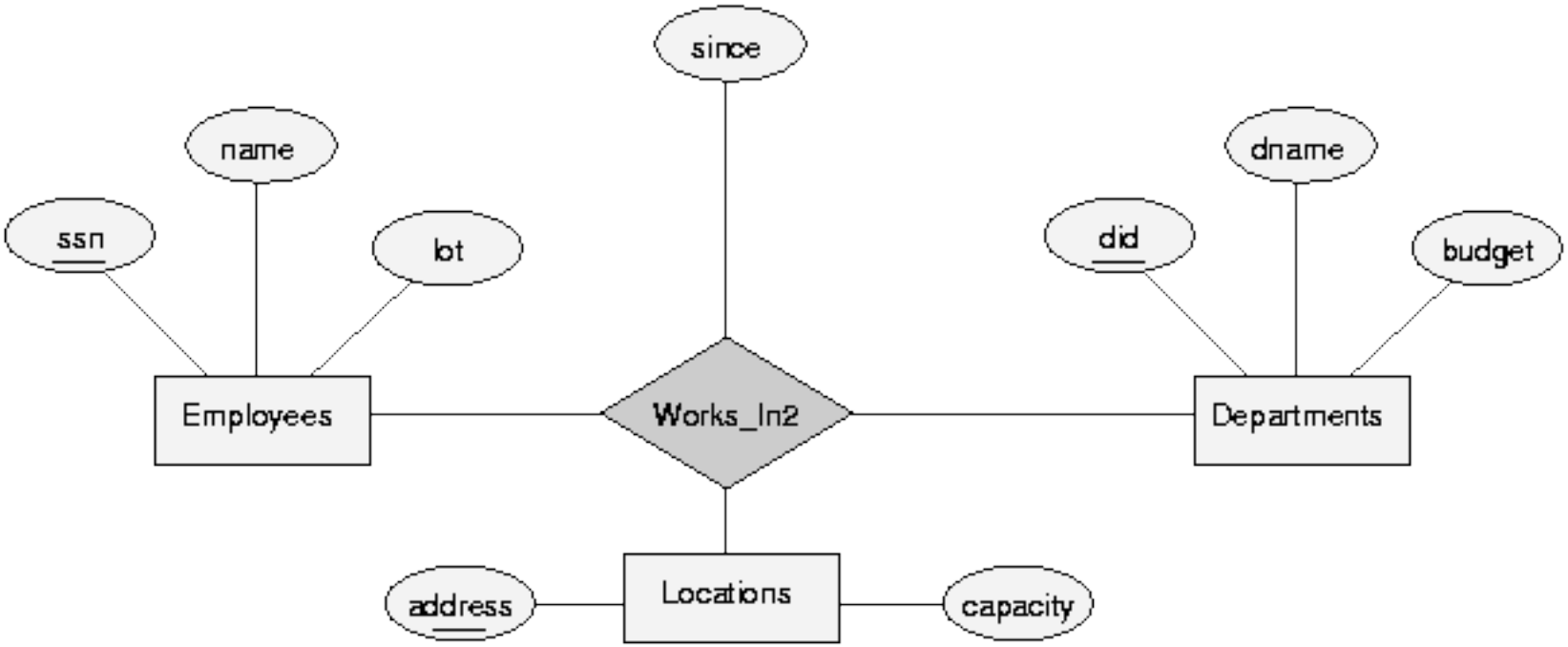
- In ER-diagrams relation-types are represented by a diamond-shape containing the name, connected to the participating entity-types

Relation-Types



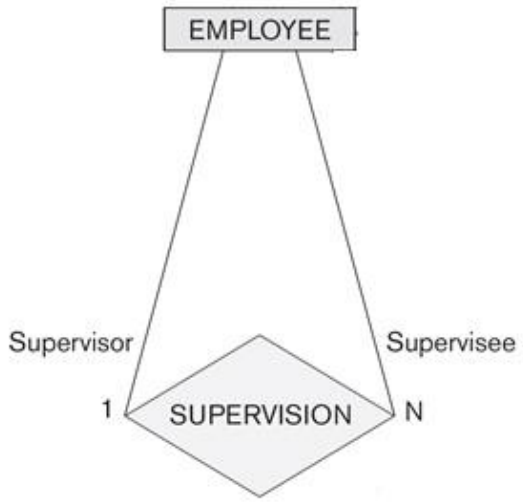
Degree of relation-types

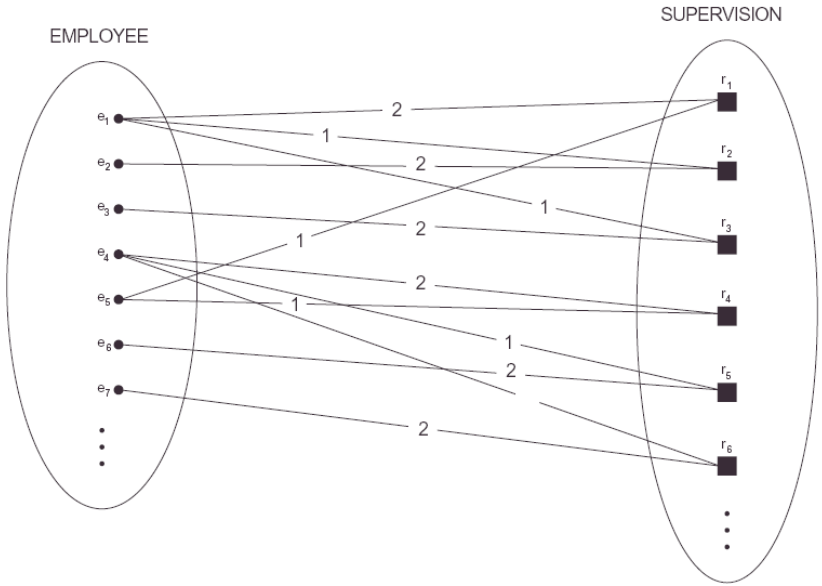
- **Degree** = number of participations by entity-types
  - binary = degree 2
  - ternary = degree 3
- Every degree (> 1) can occur
- In most cases the degree is 2


since
name
dname
ssn
lot
did
budget
Employees
Works_In2
Departments
address
Locations
capacity

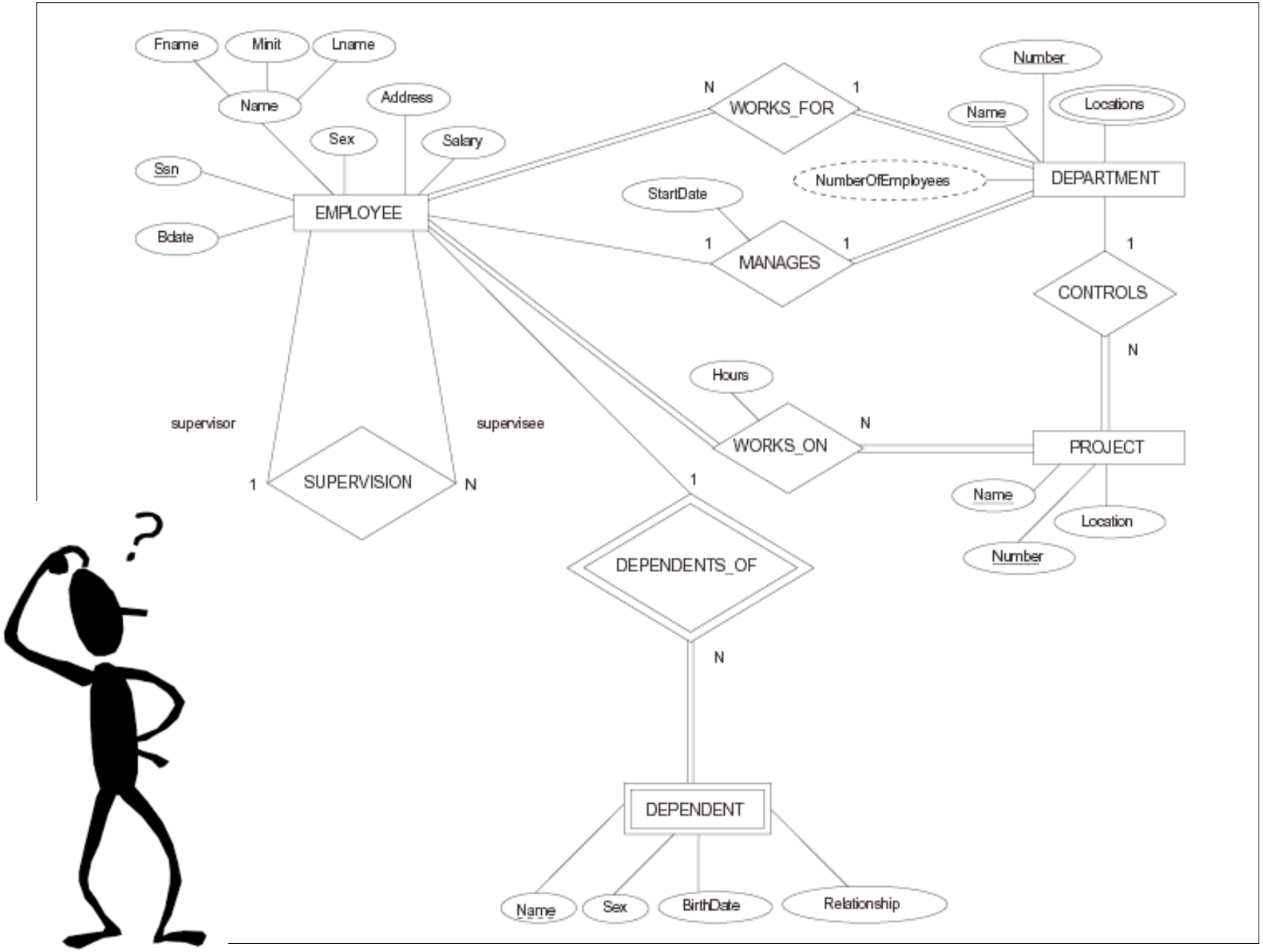
Recursive relation-types

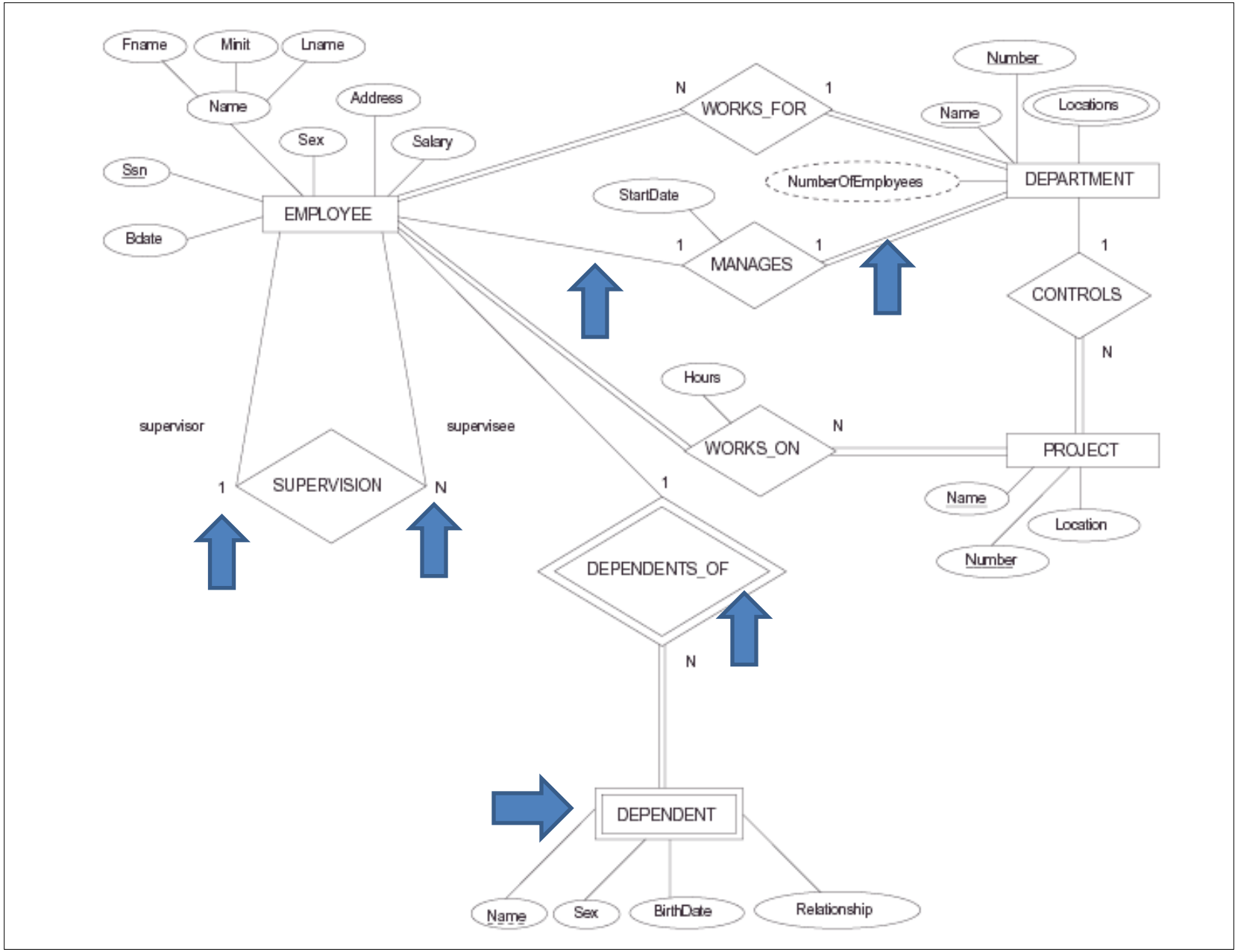
- **Roles**: each entity-type plays a *role* in a relation-type. This role can be mentioned explicitly in a diagram.
- An entity-type can play multiple roles simultaneously in a relation-type:
  - recursive relation-type
- In that case it **must** be mentioned what the roles are of such an entity





Employee plays two roles: supervisor (2) and supervisee (1)

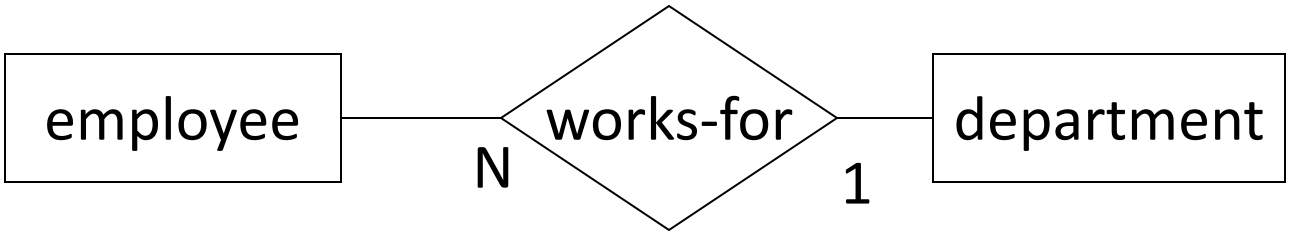

Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
Hours
supervisor
supervisee
WORKS_ON
N
PROJECT
1
SUPERVISION
N
Name
Location
Number
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship


Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
Hours
supervisor
supervisee
1
SUPERVISION
N
N
WORKS_ON
PROJECT
Name
Location
Number
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship

Cardinality ratio

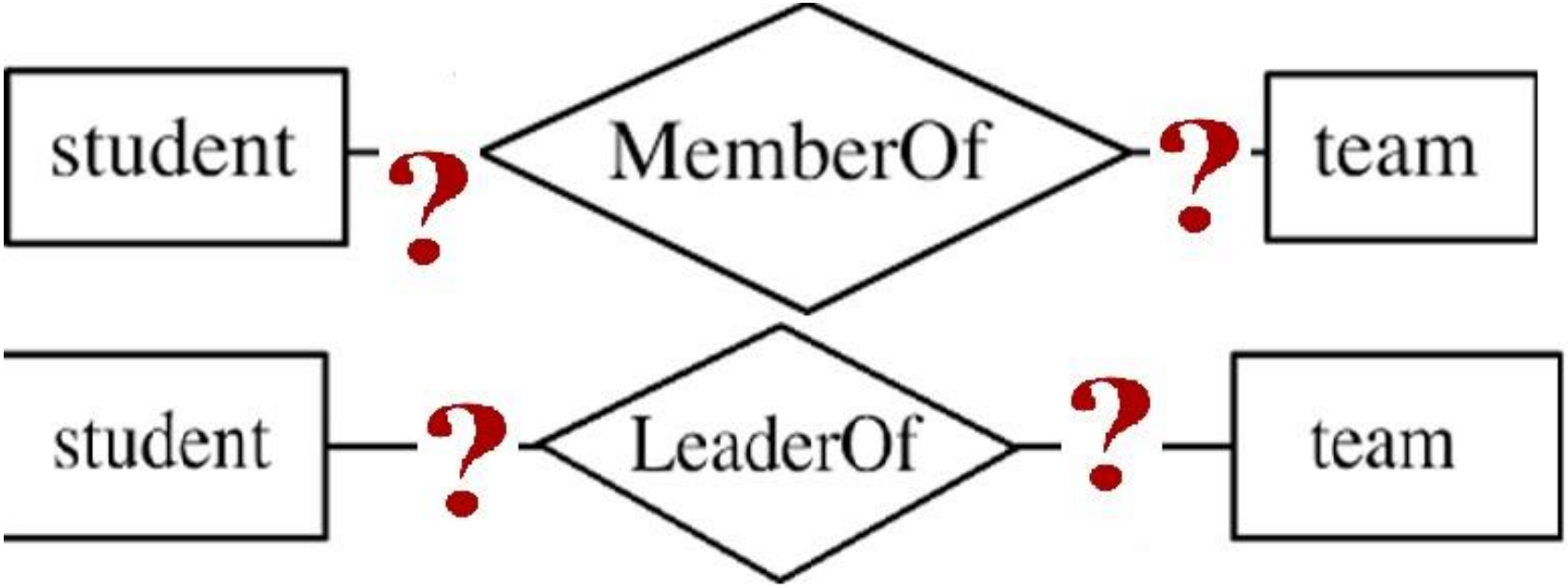
- To how many relation-instances can each entity participate?
  - The ratio for university:faculty is 1:N
    (a faculty can only belong to one university, but a university can have more than one faculty)
- Possible ratios are: 1:1, 1:N, N:1, N:M
- These values are written next to the diamonds in the diagram

Cardinality ratio

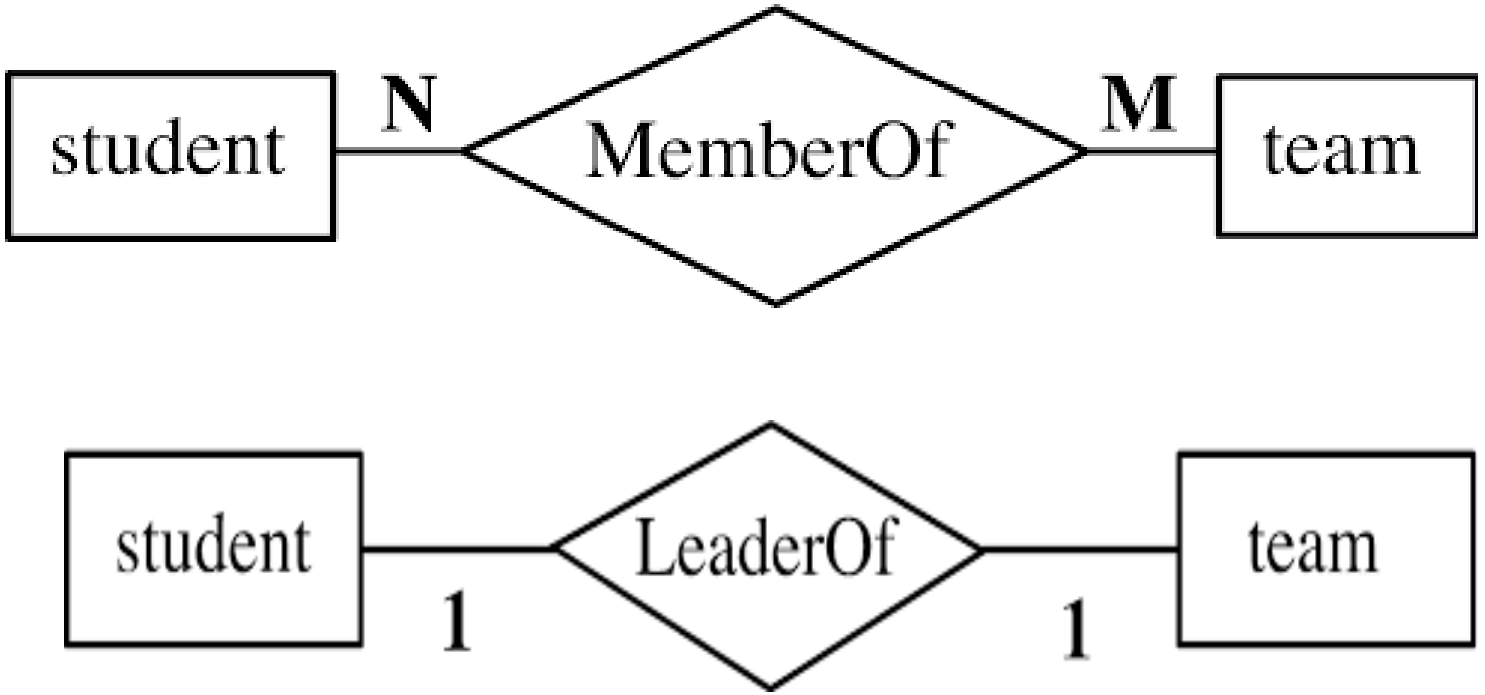


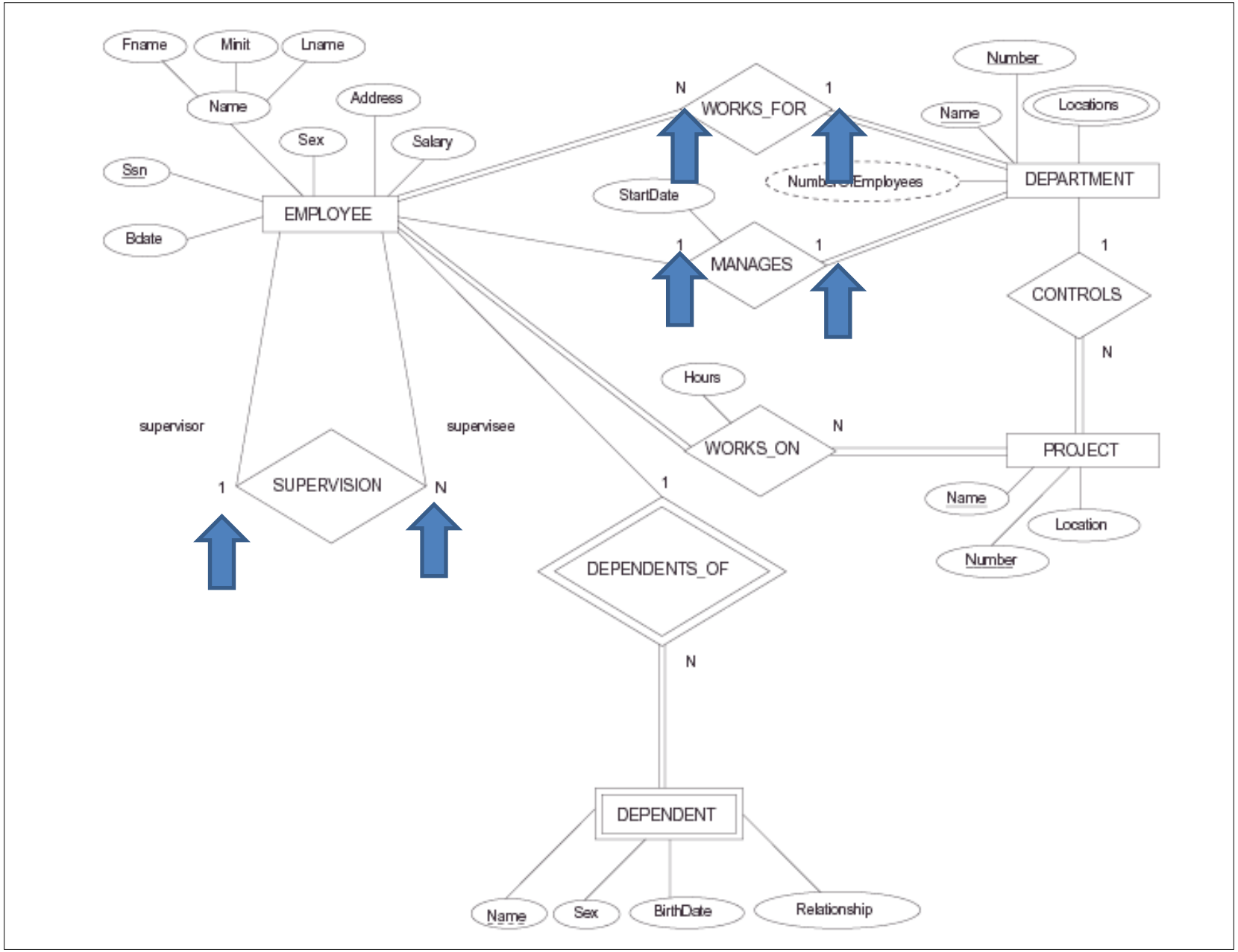
Be aware: in some books the 1 and N are swapped!
Always carefully check what these numbers mean.
In our convention, we read preferentially from left-to-right or top-to-bottom.

Cardinality ratio (More Samples)



Cardinality ratio (More Samples)

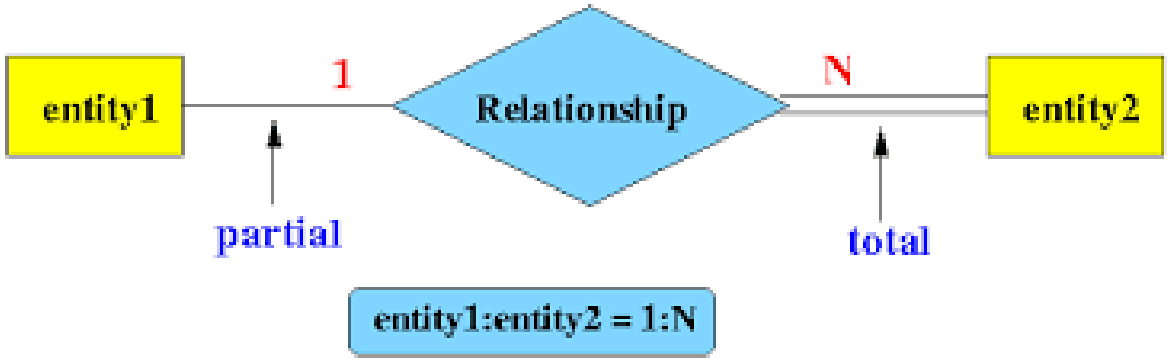



Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
StartDate
NumberOfEmployees
1
MANAGES
1
1
CONTROLS
N
Hours
supervisor
supervisee
1
SUPERVISION
N
N
WORKS_ON
PROJECT
Name
Location
Number
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship

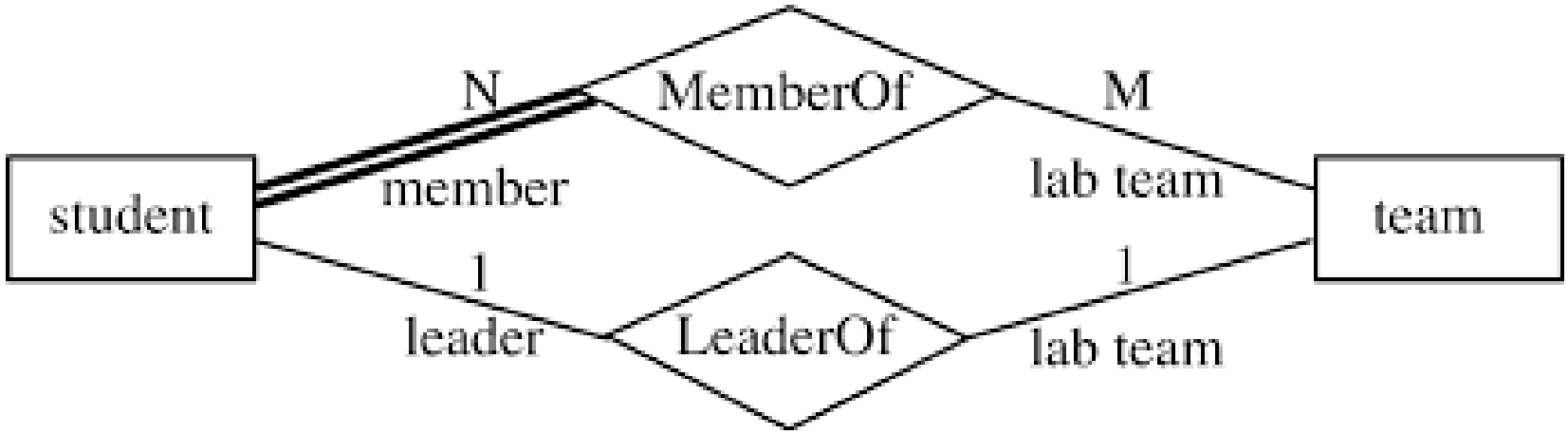
Participation restriction

- An entity-type participates **totally** if every entity of that type participates in at least one relation-instance
- If not, the participation is **partial**
- Totality is indicated by a **double line** in the diagram
- Cardinality and participation form the *structural restrictions on relation-types*

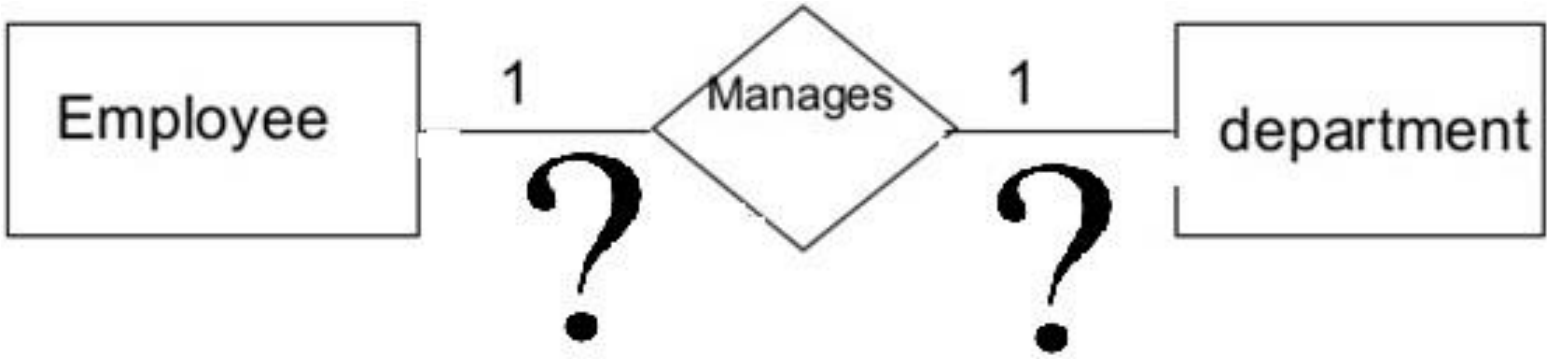
Partial/Total Participation



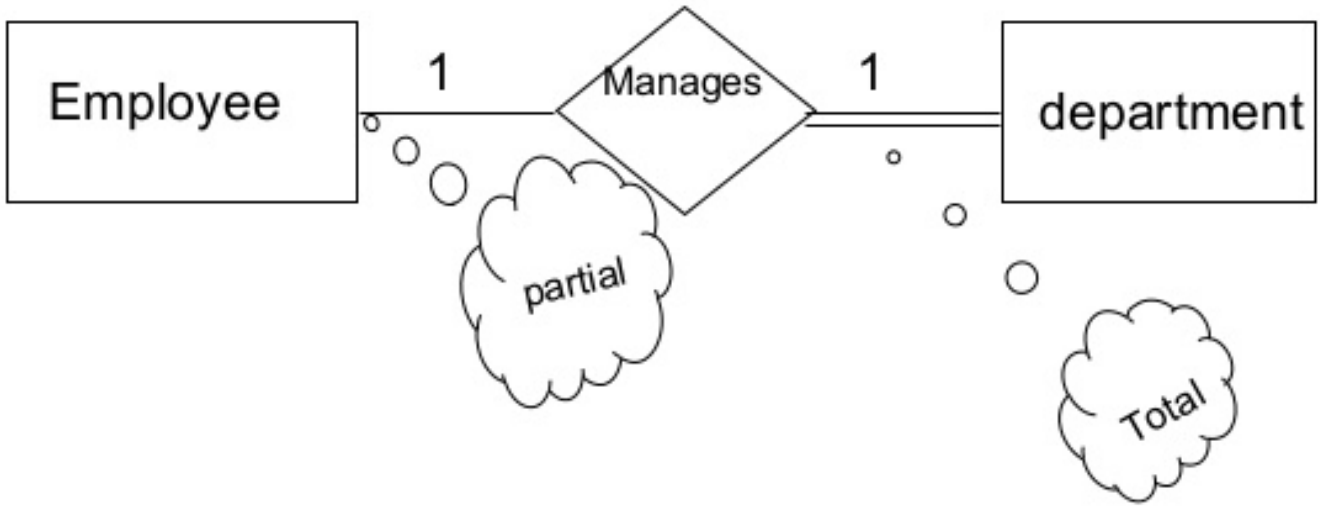
Total/Partial Participation Example 1



Total/Partial Participation Example 2

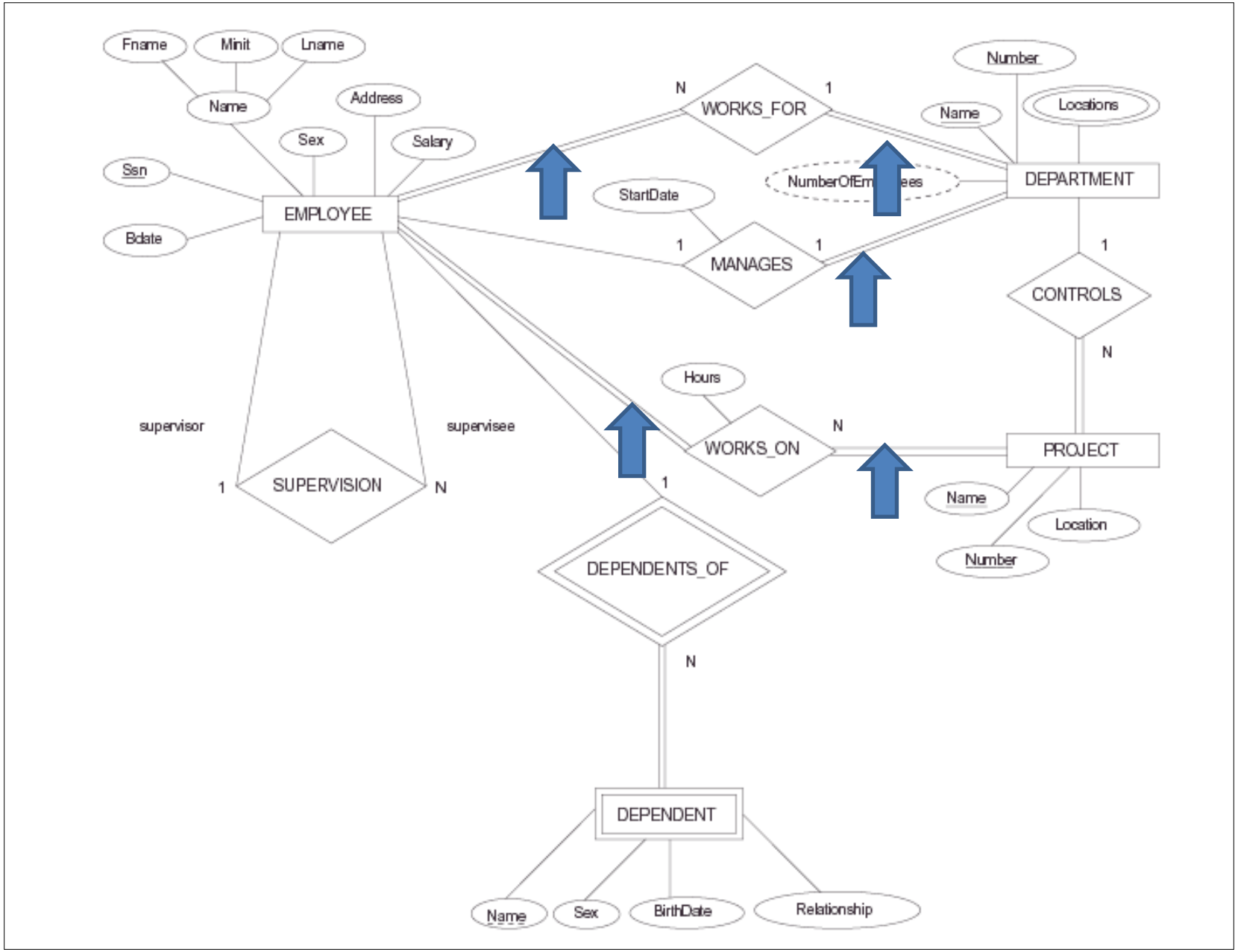


Total/Partial Participation Example 2

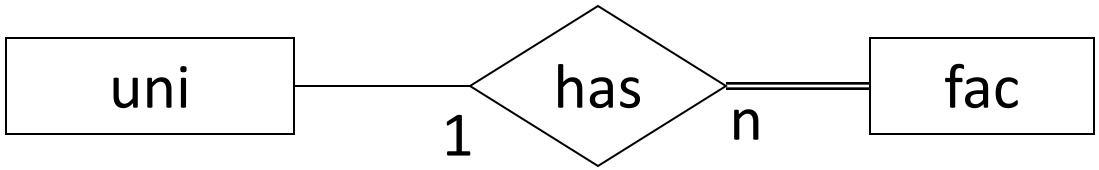


Total/Partial Participation Example 3

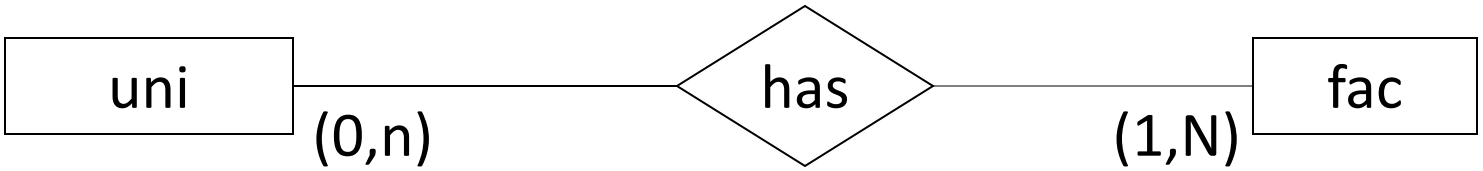
Total/Partial Participation Example 3


Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
Hours
supervisor
supervisee
1
SUPERVISION
N
WORKS_ON
N
PROJECT
Name
Location
Number
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship

Alternative Notation

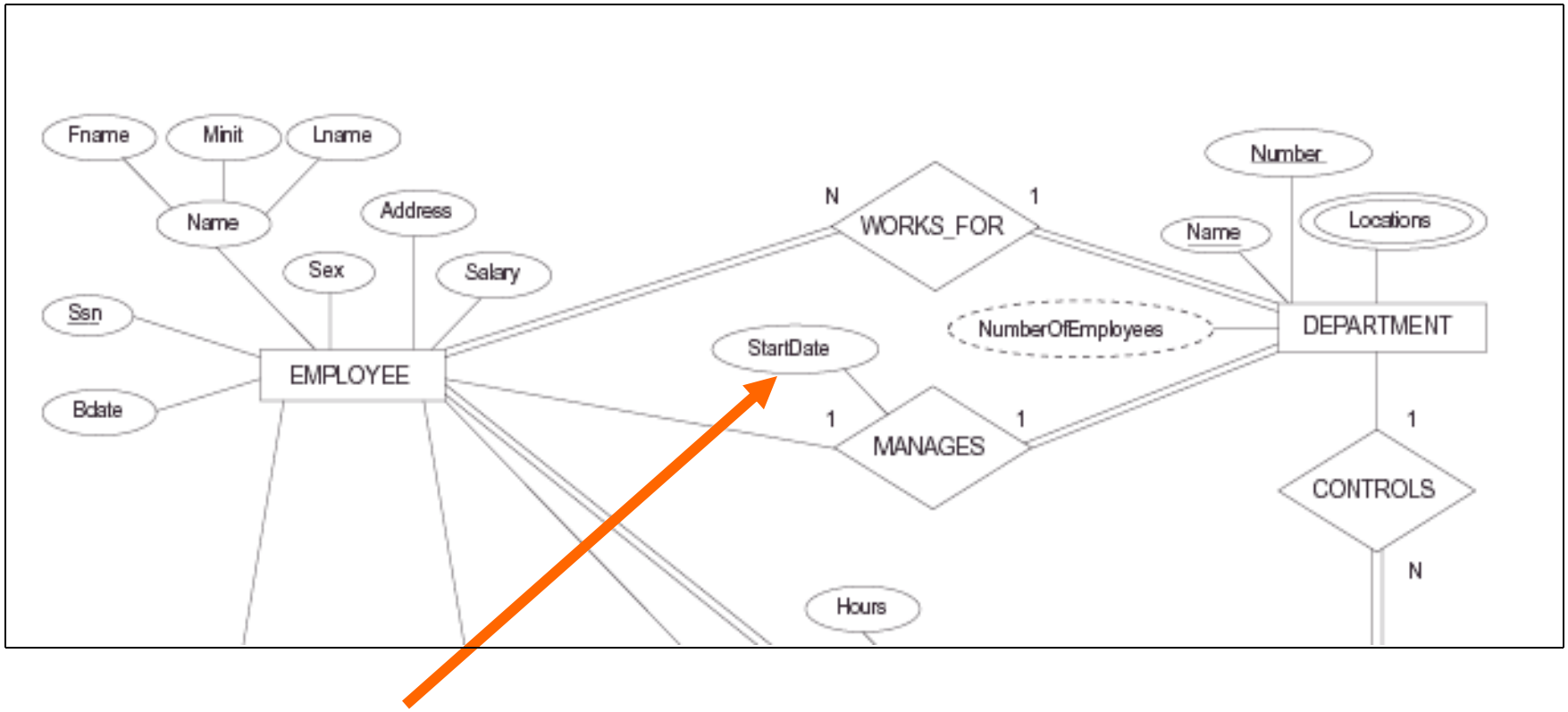


- Write at the entity-type the *minimum* and *maximum* number of participations per entity allowed:



Attributes of relation-types

- Also relation-types can have attributes
  - Student id122773 participates_in databases *in the academic year 2012/2013*
- Indicated by ellipses connected to the diamond
- There are no keys for relation-types
- In 1:1 or 1:N relations: the attributes can move to an entity-type



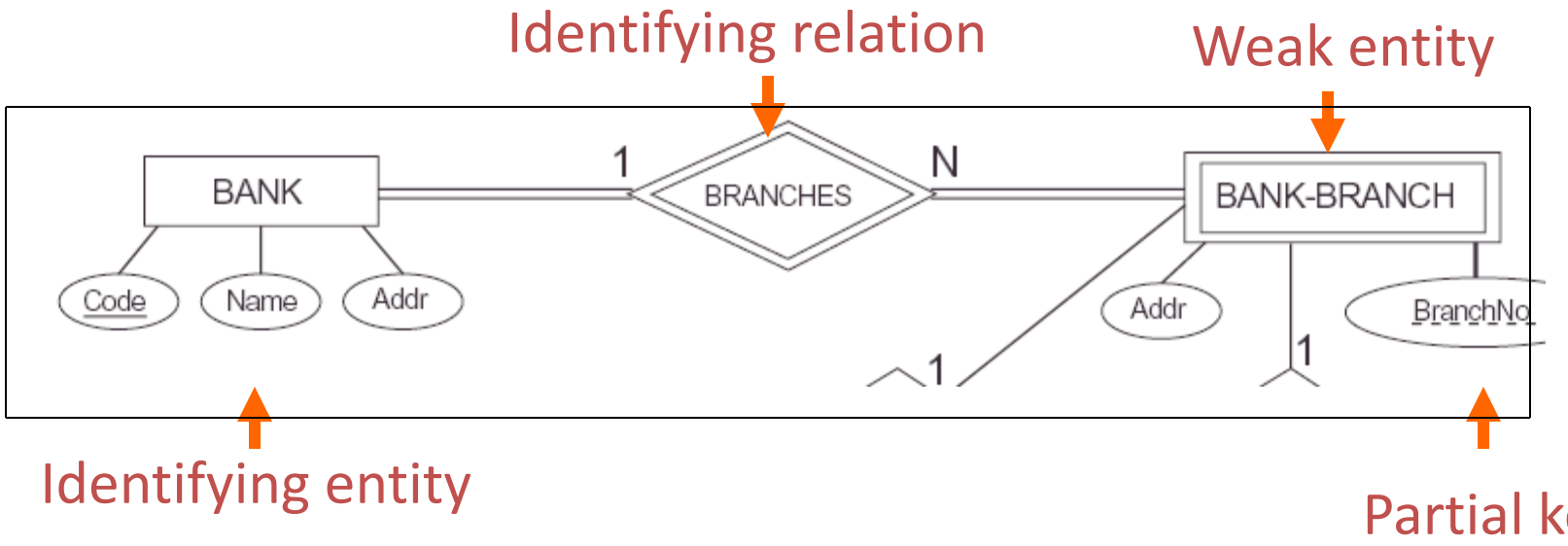
May move to EMPLOYEE or to DEPARTMENT

Weak Entity Types

- Do not have key attributes of their own
  - Identified by being related to specific entities from another entity type
- **Identifying relationship**
  - Relates a **weak** entity type to its **owner**
- Always has a **total** participation constraint

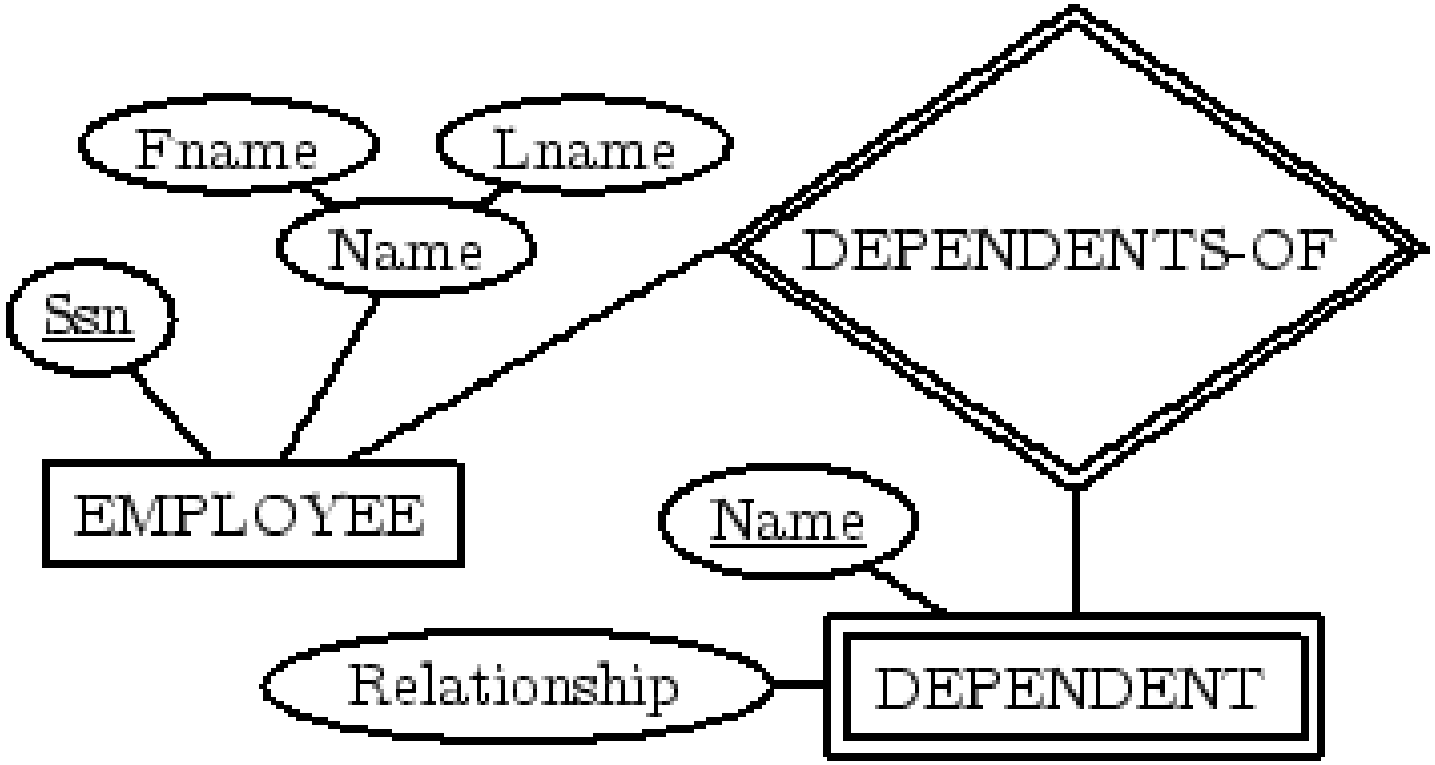
Weak entity-types

- Indicated by double lines for the weak entity type (rectangle) and the identifying relation type (diamond)
  - partial keys are underlined with a broken line

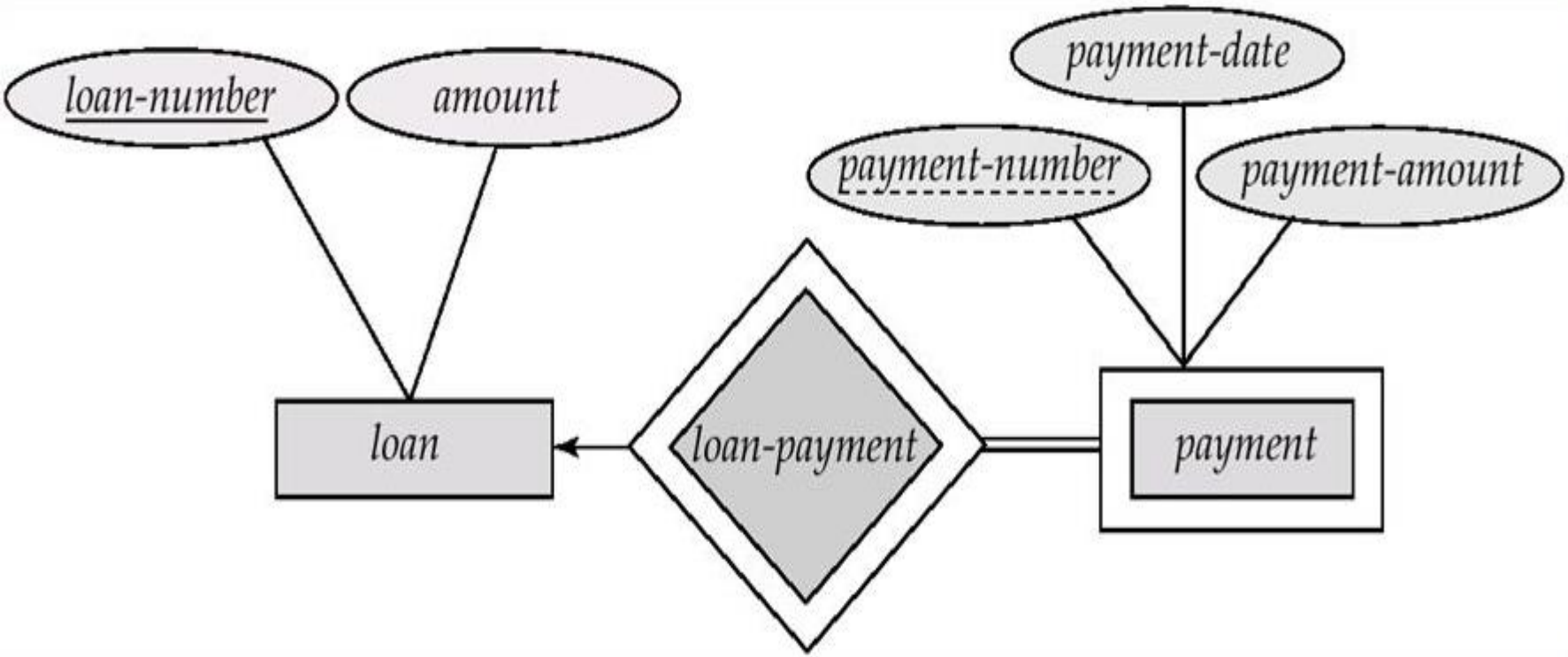


New key: bank.code + bank-branch.branchno

Weak Entity Samples



Weak Entity Samples



Summary of notation in ER diagrams

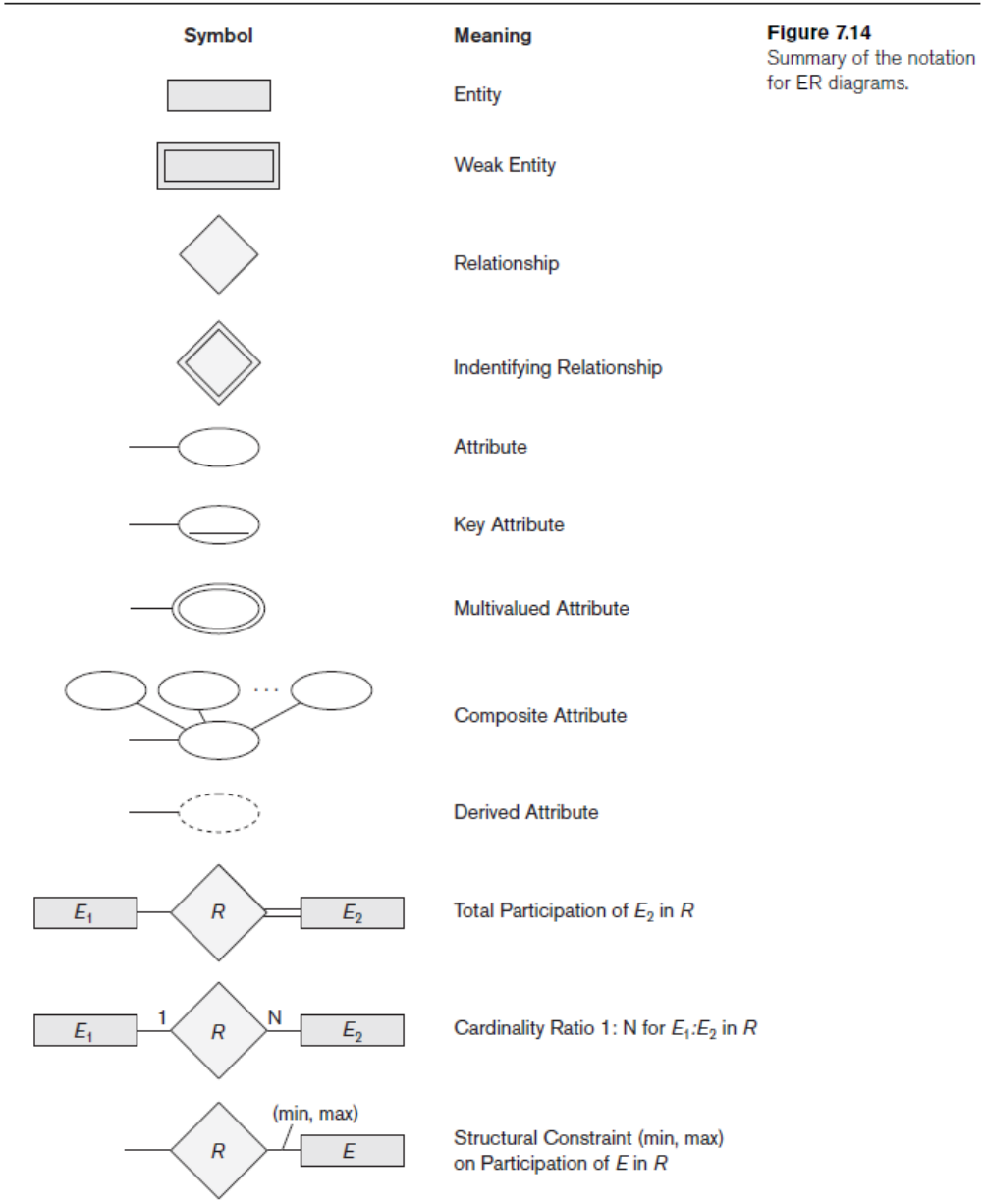
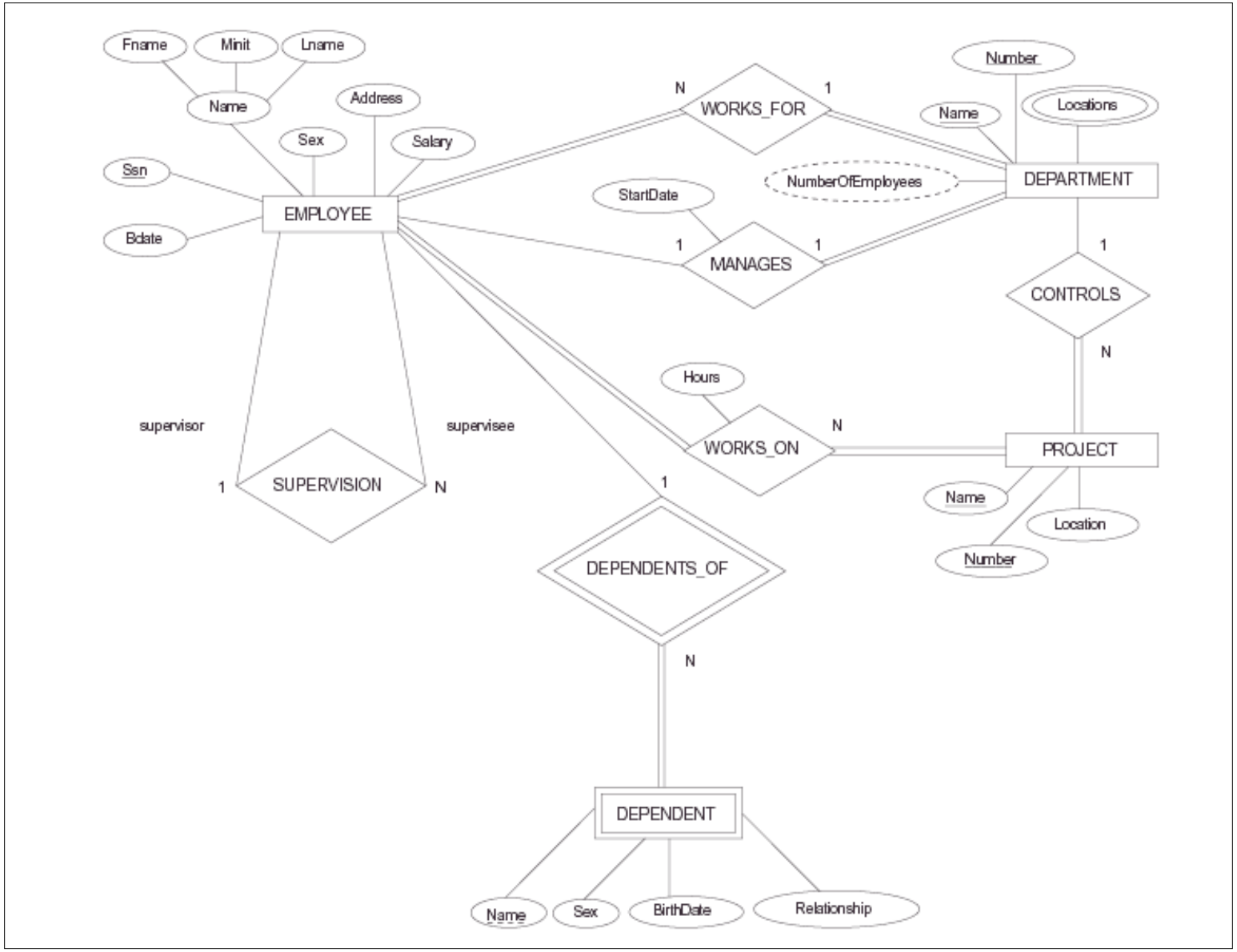


Figure 7.14
Summary of the notation for ER diagrams.

Designing a complete ER-diagram

- Iterative task: designing, going back to the users, re-designing, etc.
- Questions might be:
  - Should initial attributes better by modeled as relation types?
  - What are the cardinalities of the relations?
  - Should relations be total?
  - Should entities be split into identifying and weak entities?
- This leads to a final ER-diagram

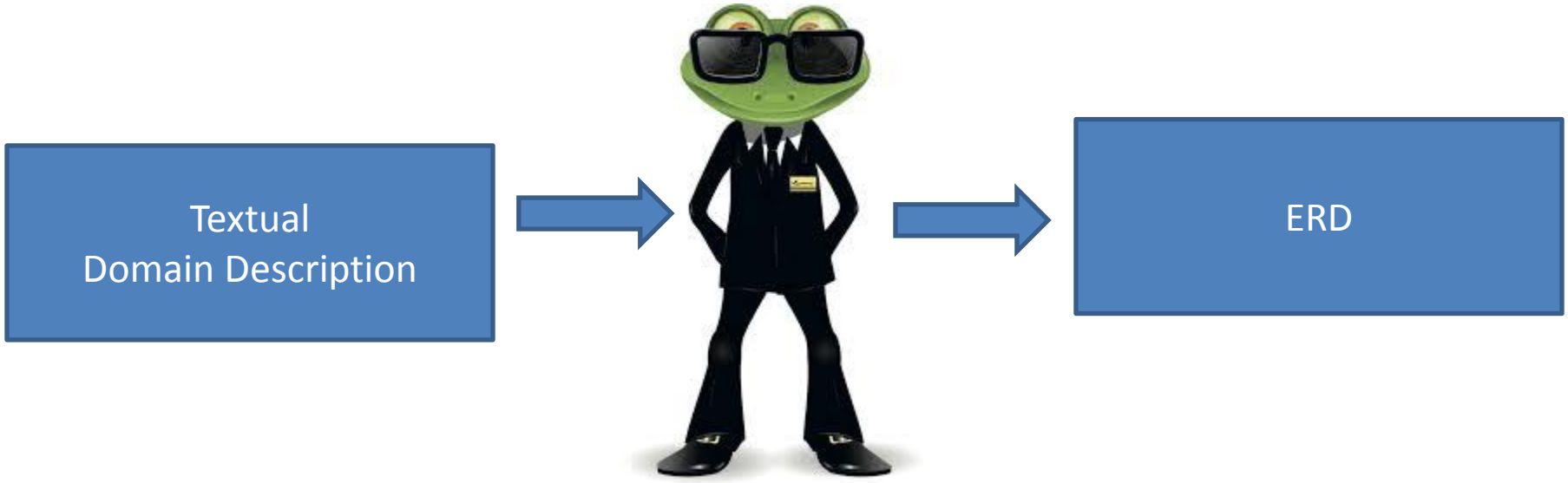

Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
Hours
WORKS_ON
N
PROJECT
Name
Location
Number
supervisor
supervisee
1
SUPERVISION
N
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship

Data Modeling using the Entity Relationship (ER) Model

- High-Level Conceptual Data Models
- Entity Relationship Model and its elements
- From text to ER-schema

From text to ER-schema

- Often, you will create an ER diagram based on a textual description. There are some rules-of-thumb that can help you in this task



From text to ER-schema

- Noun: entity-type
  - In our **university**, **student**s follow **course**s that are given by a **teacher**.
- Verb that connects nouns: relation type
  - In our university, students **follow** courses that are **given** by a teacher.
- Adjective: attribute at entity-type
  - The **large** shop sells **expensive** cars.

From text to ER-schema

- Adverb: attribute at relation-type
  - The student **successfully** finishes the course.
  - The car is rent by the client **for a certain period of time**.

From text to ER-schema

- When you are done, inspect the diagram and improve it where possible:
  - Identify possible keys
  - Identify weak entity-types
  - Entity-types without attributes can better become attributes of a relation or another entity
  - Identify and remove redundant relations
  - Remove entity-types without relations
  - Remove entity-types with only a single entity

From text to ER-schema

- Describe the domain of all attributes
- Write down all restrictions that cannot be represented in the diagram
- Describe all operations on the data that are needed for the database application (data entry, removal, etc.)

Other restrictions

- The ER-diagram and schema cannot represent all restrictions from the mini world
- Those restrictions, however, can be extremely important when building the database application
- Write down these restrictions as a note to the diagram and schema

Summary

- Basic ER model concepts of entities and their attributes
  - Different types of attributes
  - Structural constraints on relationships

Quiz/part1

- A university database contains information about professors (identied by social security number, or SSN) and courses (identied by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram that describes it (assuming that no further constraints hold).
- 1. Professors can teach the same course in several semesters, and each offering must be recorded.
- 2. Professors can teach the same course in several semesters, and only the most recent such offering needs to be recorded. (Assume this condition applies in all subsequent questions.)

Quiz/Part2

- 3. Every professor must teach some course.
- 4. Every professor teaches exactly one course (no more, no less).
- 5. Every professor teaches exactly one course (no more, no less), and every course must be taught by some professor.
- 6. Now suppose that certain courses can be taught by a team of professors jointly, but it is possible that no one professor in a team can teach the course.