Entity-Relationship Model

Chapter 2

Contents

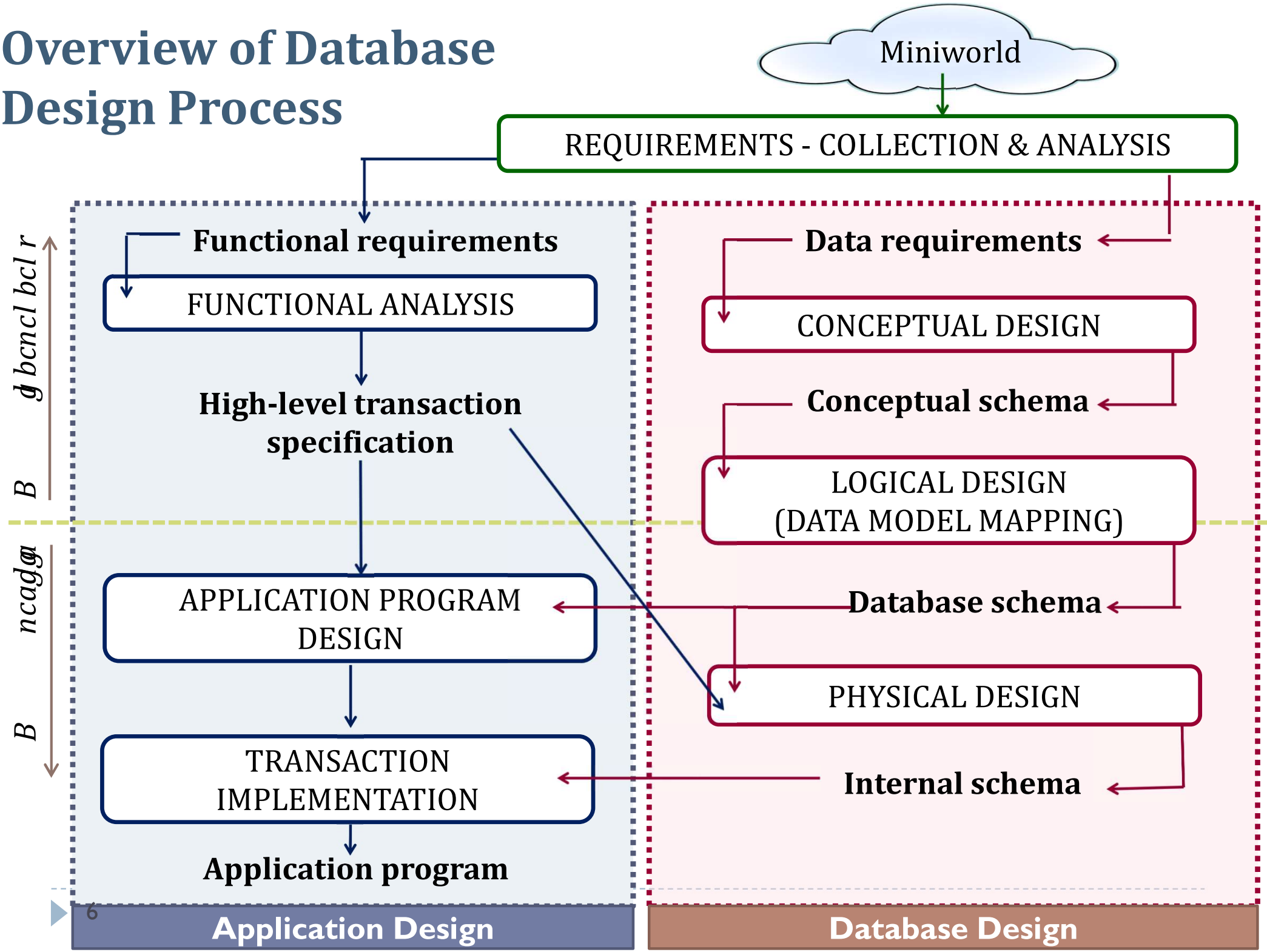
Contents

Contents

Overview of Database Design Process

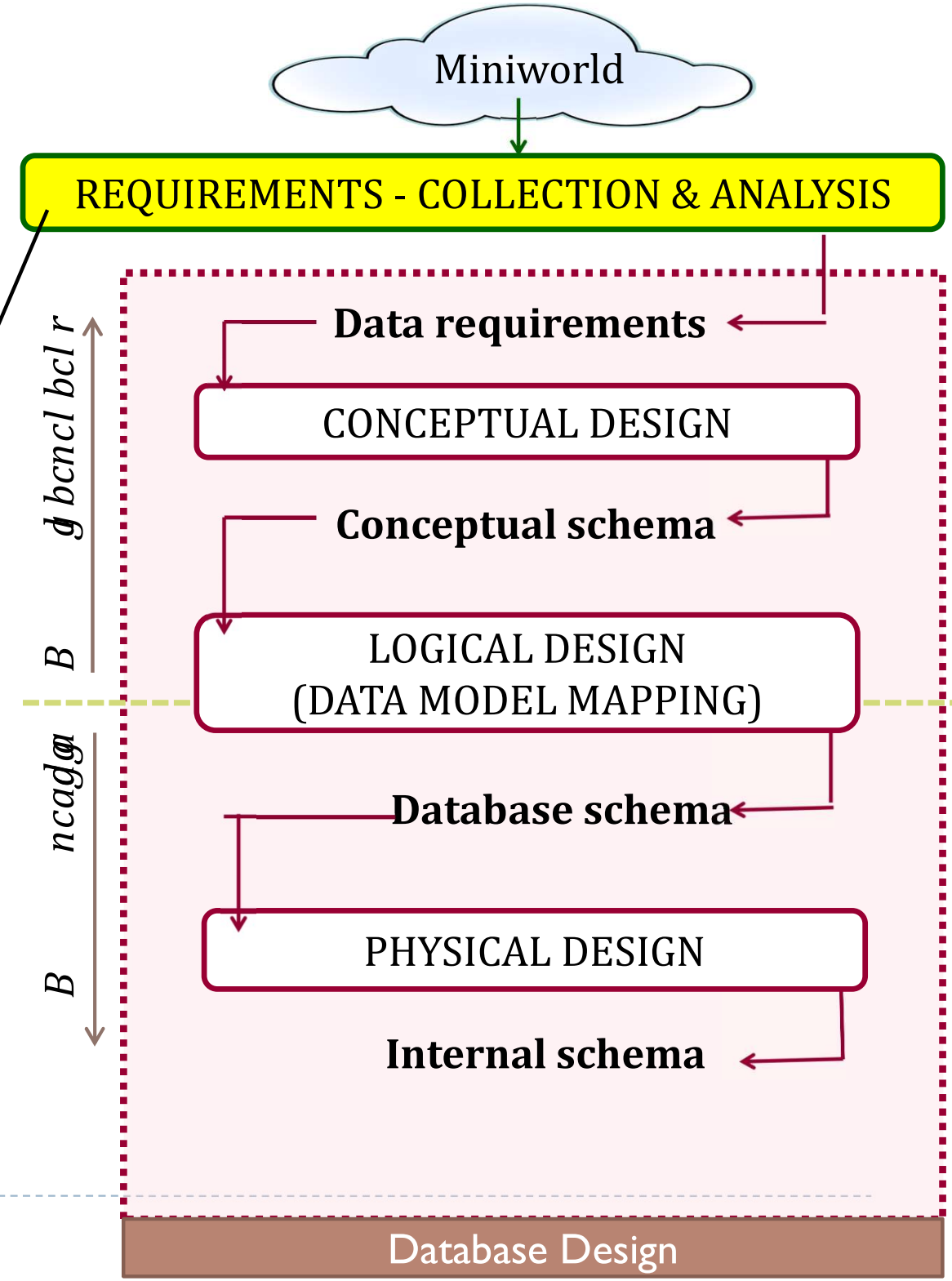
- **Database design**
  - To design the conceptual schema for a database application
- **Applications design**
  - Focus on the programs and interfaces that access the database
  - Generally considered part of software engineering

Overview of Database Design Process



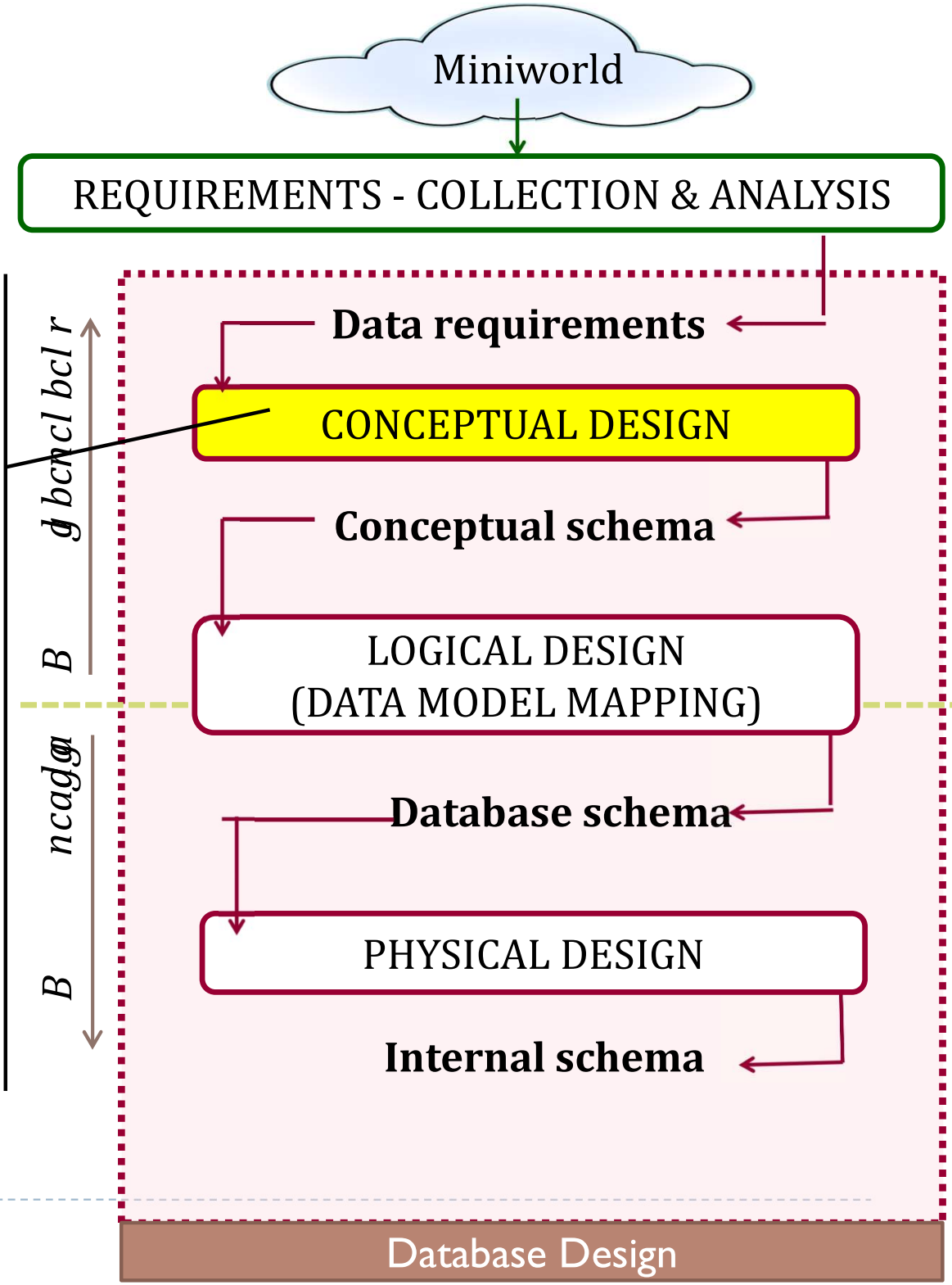
Overview of Database Design Process

- Interview prospective database users
- Result:
  - ✓ Data requirements
  - ✓ Functional requirements



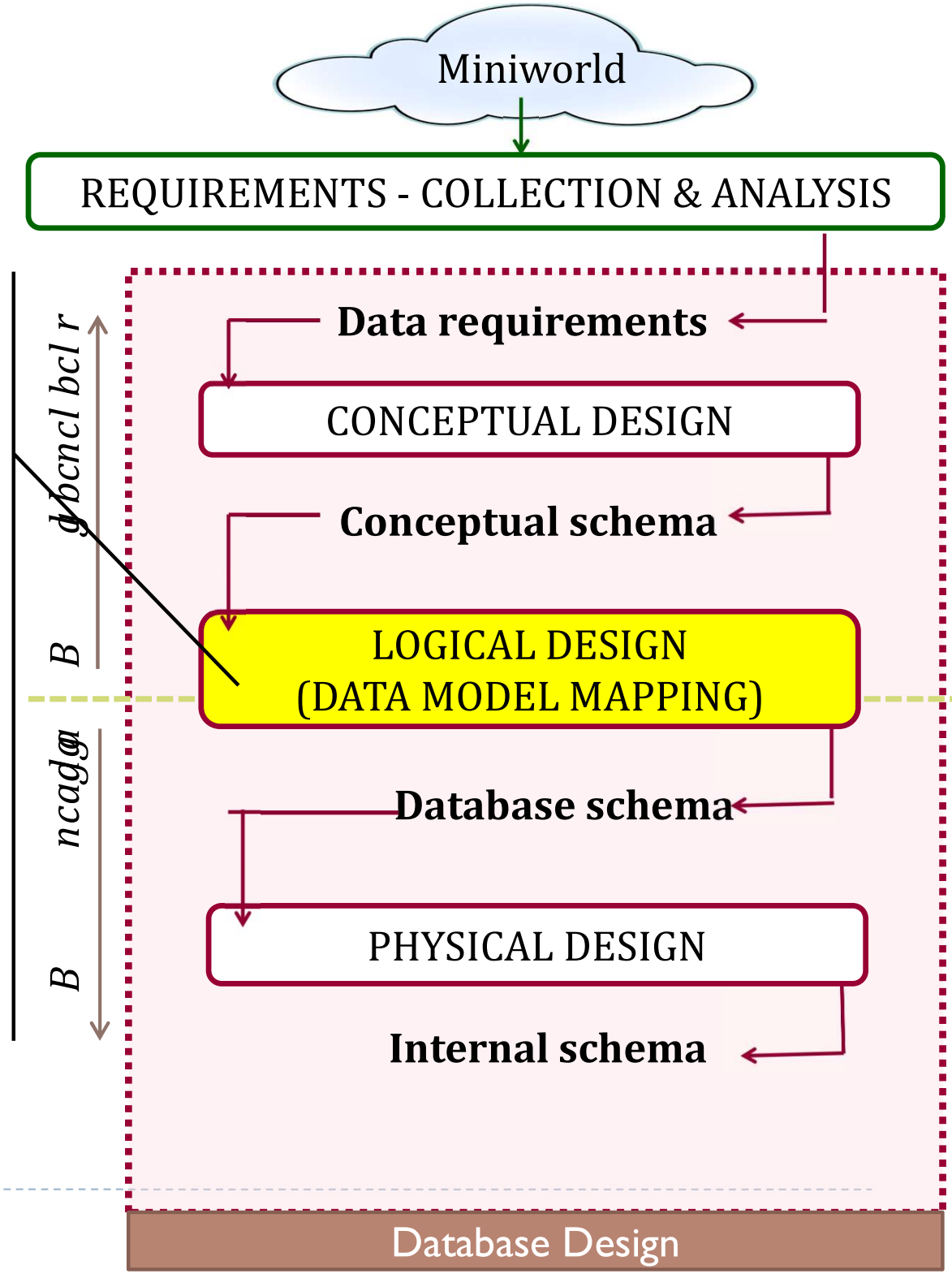
Overview of Database Design Process

- Create a conceptual schema using a high-level conceptual data model **(Entity-Relationship model)**
- Descriptions of entity types, relationships, and constraints
- **Independent** of storage and implementation details.



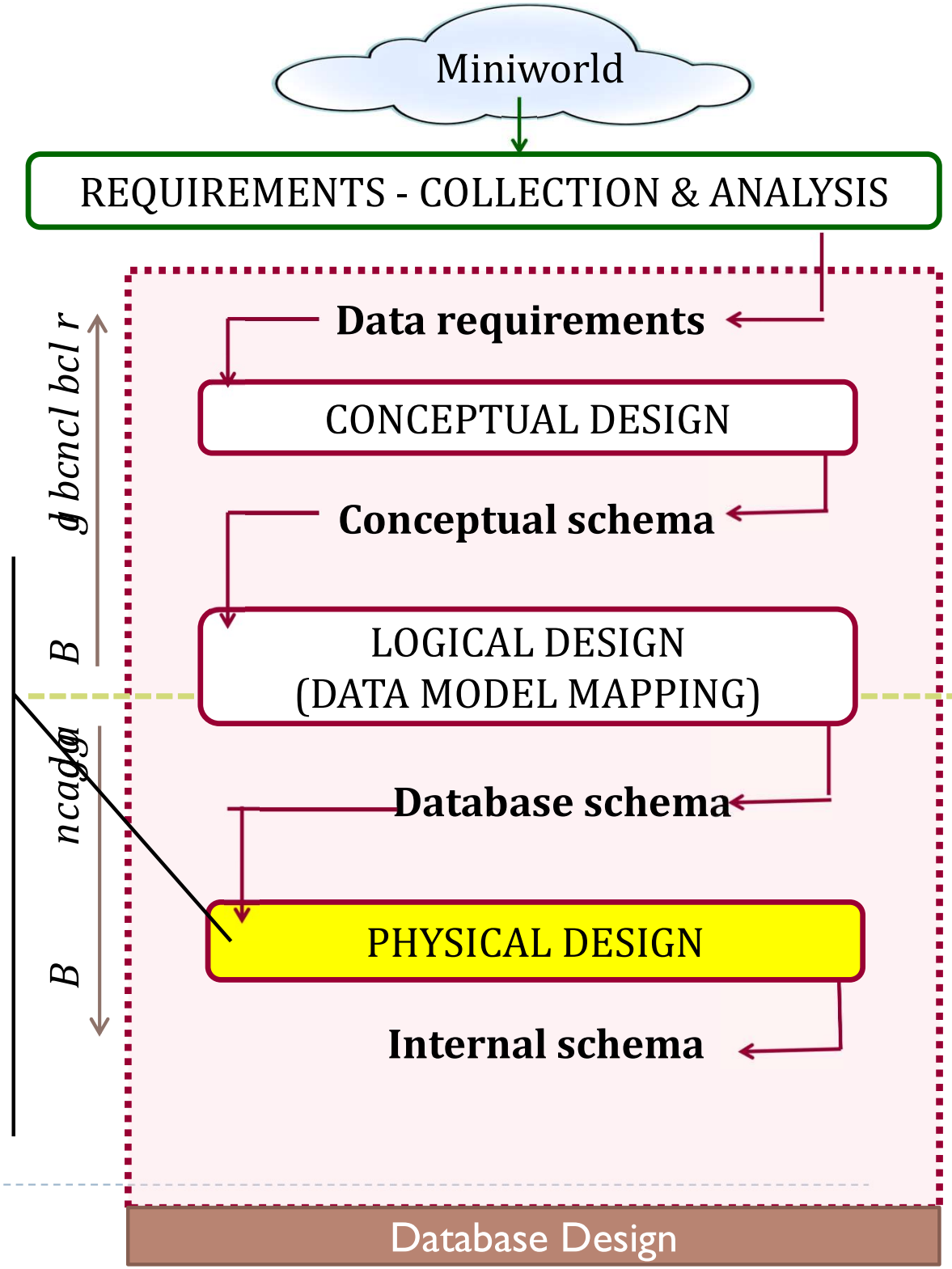
Overview of Database Design Process

- Create a **database schema** in implementation data model of a commercial DBMS
- **Data model mapping** is often automated or semi-automated within the database design tool.



Overview of Database Design Process

- Specify internal storage structures, file organizations, indexes, access paths, and physical design parameters for the database files.

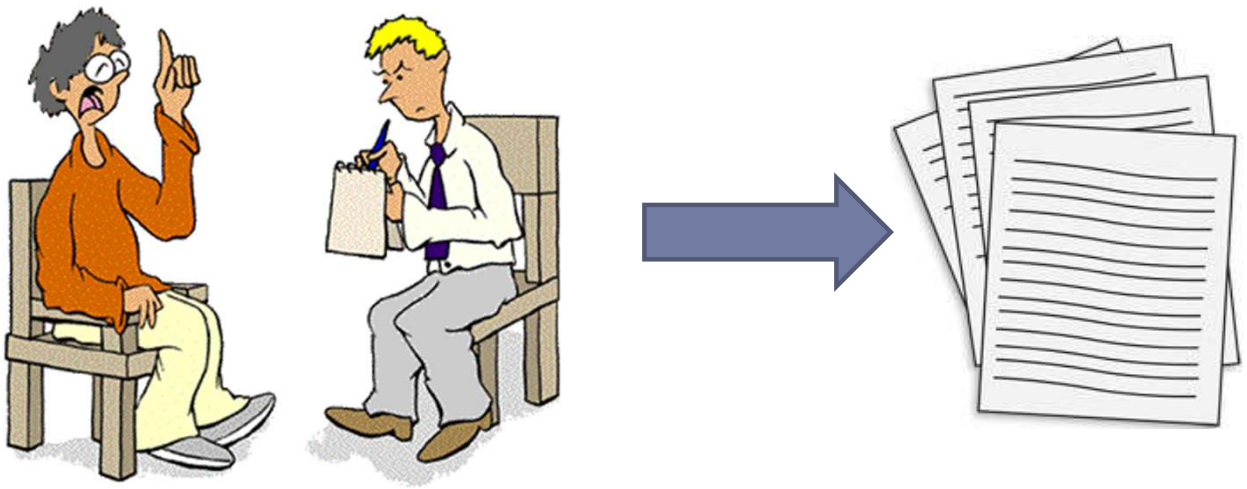


Contents

A Sample Database Application

- Design a database for a **COMPANY** that keeps track of employees, departments, and projects

  ➔ REQUIREMENTS - COLLECTION & ANALYSIS



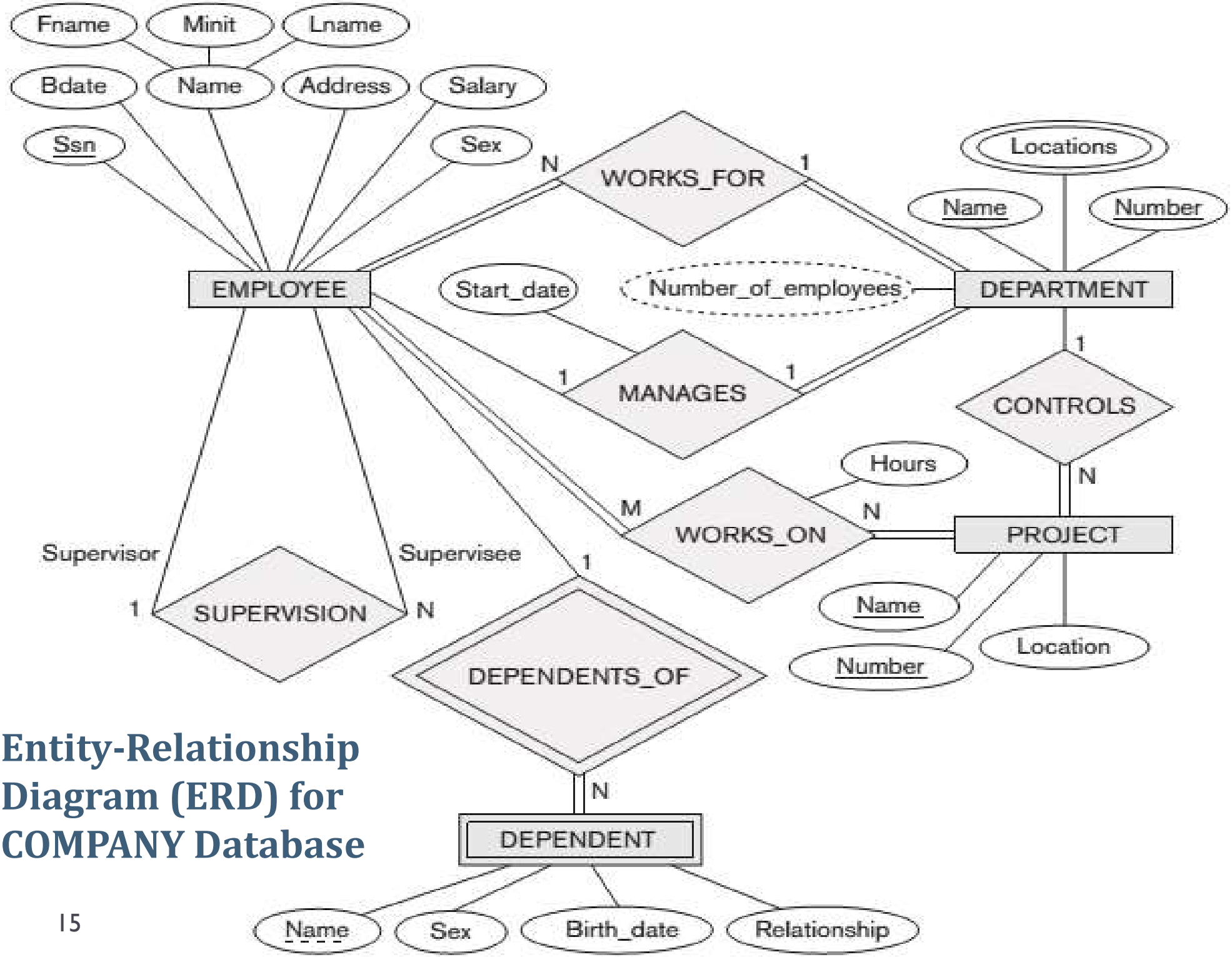
Data requirements
What data should be stored?
- Entities
- Attributes
- Relationships
- Constraints

A Sample Database Application

- The **COMPANY** database: keeps track of employees, departments, and projects.
- The company is organized into **DEPARTMENTs**. Each department has a unique name, a unique number, and a particular employee who *k l e c* the department. We keep track of the start date when that employee began managing the department. A department may have several locations.
- A department *aml rpmj* a number of **PROJECTs**, each of which has a unique name, a unique number, and a single location.

A Sample Database Application

- We store **EMPLOYEE**'s name, Social Security number, address, salary, sex, and birth date. An employee is *g l cb rmml c bcn prk cl r*, but may *u mpi ml ct c p j npmhcar* , which are not necessarily controlled by the same department. We keep track of the current number of hours per week that an employee works on each project. We also keep track of the direct *sncpt gmp* of each employee.
- We want to keep track of the **DEPENDENTs** of each employee, including first name, sex, birth date, and relationship to the employee.



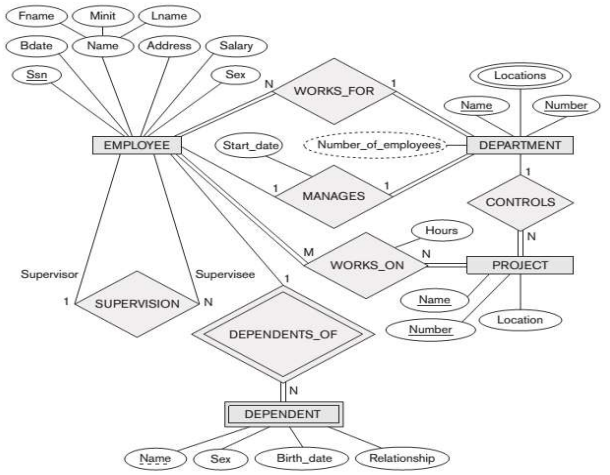
Entity-Relationship Diagram (ERD) for COMPANY Database

Contents

What is ER Model?

- Entity-Relationship (ER) model
  - Popular high-level conceptual data model
  - A logical organisation of data within a database system
- ER Diagrams (ERD):
  - Diagrammatic notation associated with the ER model
- Conceptual Design:

  Data requirements → Conceptual Schema (ERD)



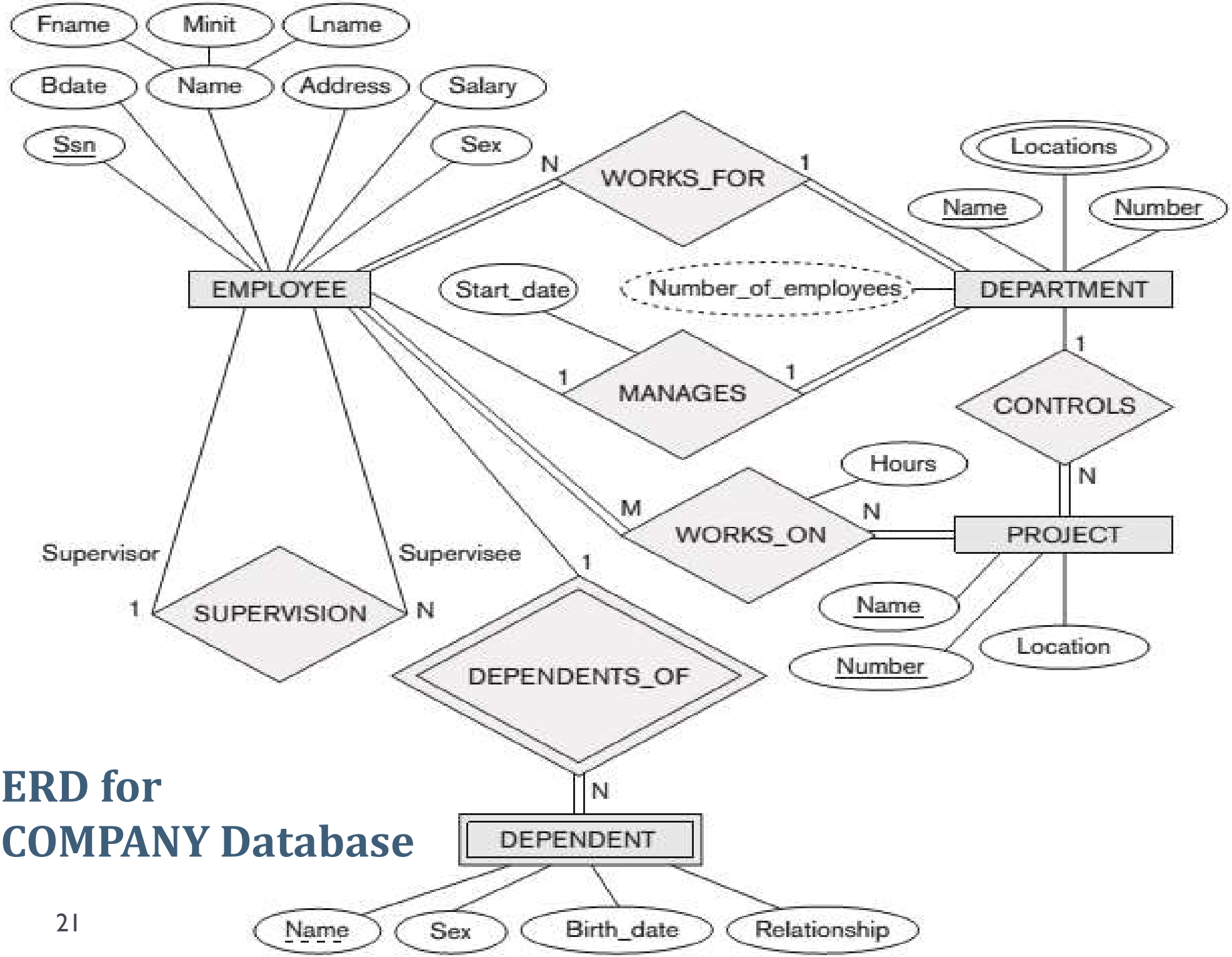
Why use ER data modelling?

- User requirements can be *ncaggcbdmpk jjw sl k gg smsjw*
- It can be *c gjw sl bcp rmmb* by ordinary users.
- It provides *l ccdcargtc pgbe c* between user requirements and logical database design and implementation
- The conceptual data model is *gj bcncl bcl r md lw n prgasj pB*
- It does *l mr gj t mjtc lwn fw ga jmp gk n jck cl r jbcr gj*

Contents

ER Model Concepts

- ER model describes data as:
  - Entities
  - Attributes
  - Relationships



ERD for COMPANY Database

Entity

- **Entity** is a thing in the real world with an independent existence.
- An entity may be an object with a *nfw gq jcvgrcl ac* (a person, a car, a house, or an employee) or an object with a *aml acnrs jcvgrcl ac* (a company, a job, or a university course)
- Examples: In a COMPANY:
  - the EMPLOYEE *mfl k gf*
  - the *Pc c paf* DEPARTMENT
  - the *NpmbsanV* PROJECT

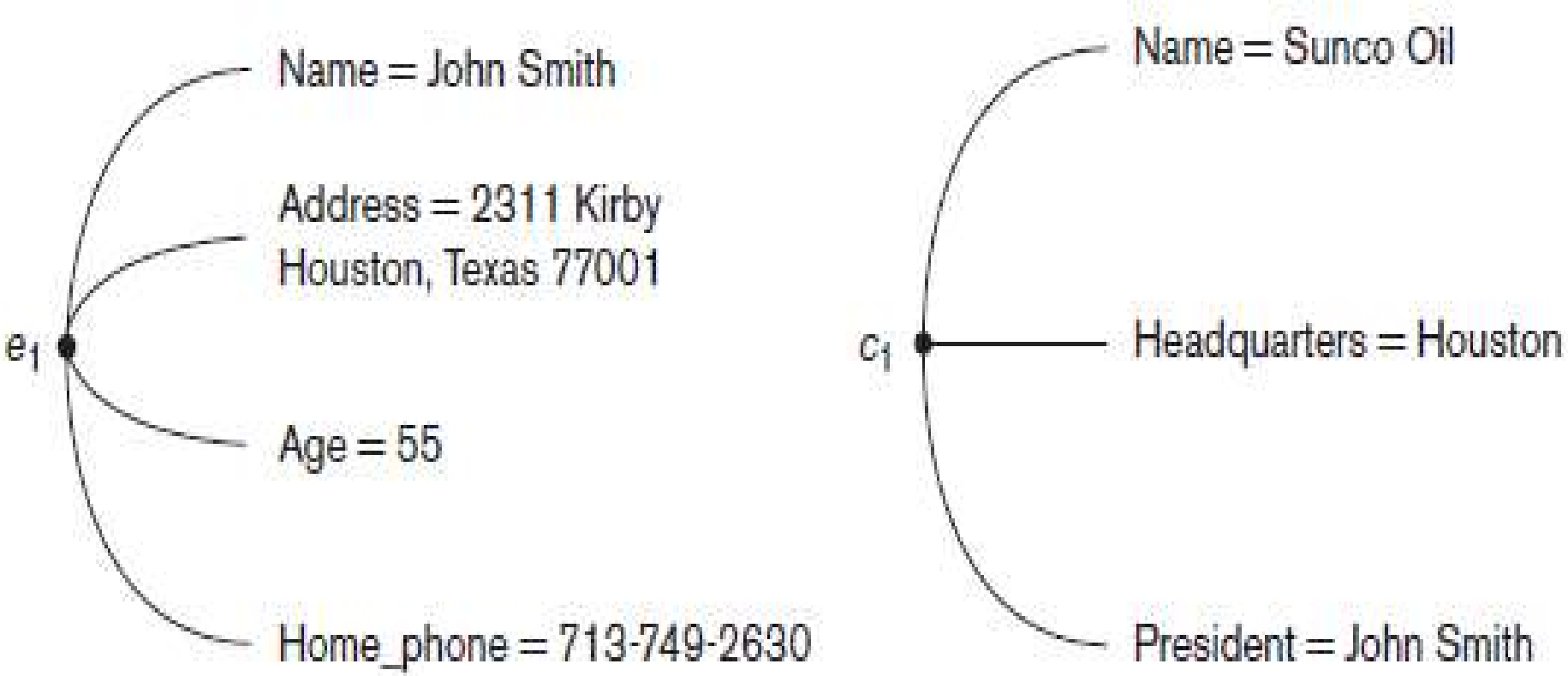
Attribute

- **Attributes** are properties described an entity.
  - Ex: an EMPLOYEE entity may have Name, SSN, Address, Sex, BirthDate
- A specific entity will have a value for each of its attributes.
- Each attribute has **a value set (or data type)** associated with it.

Types of Attributes

- *dk n jc rrpg src* has a single atomic value.
  - SSN, Sex
- *mk nm gc rrpg src* may be composed of several components.
  - Name (First name, Middle name, Last name)
- *s jrgt jscb rrpg src* has multiple values.
  - Colors of a Car {Color}, Phones of a Person {Phone}
- *Bcpgtcb rrpg src* has a value that is derivable from values of related attributes.
  - Number of students in a class
- *mk n jcv rrpg src* is a combination of composite and multivalued attributes.

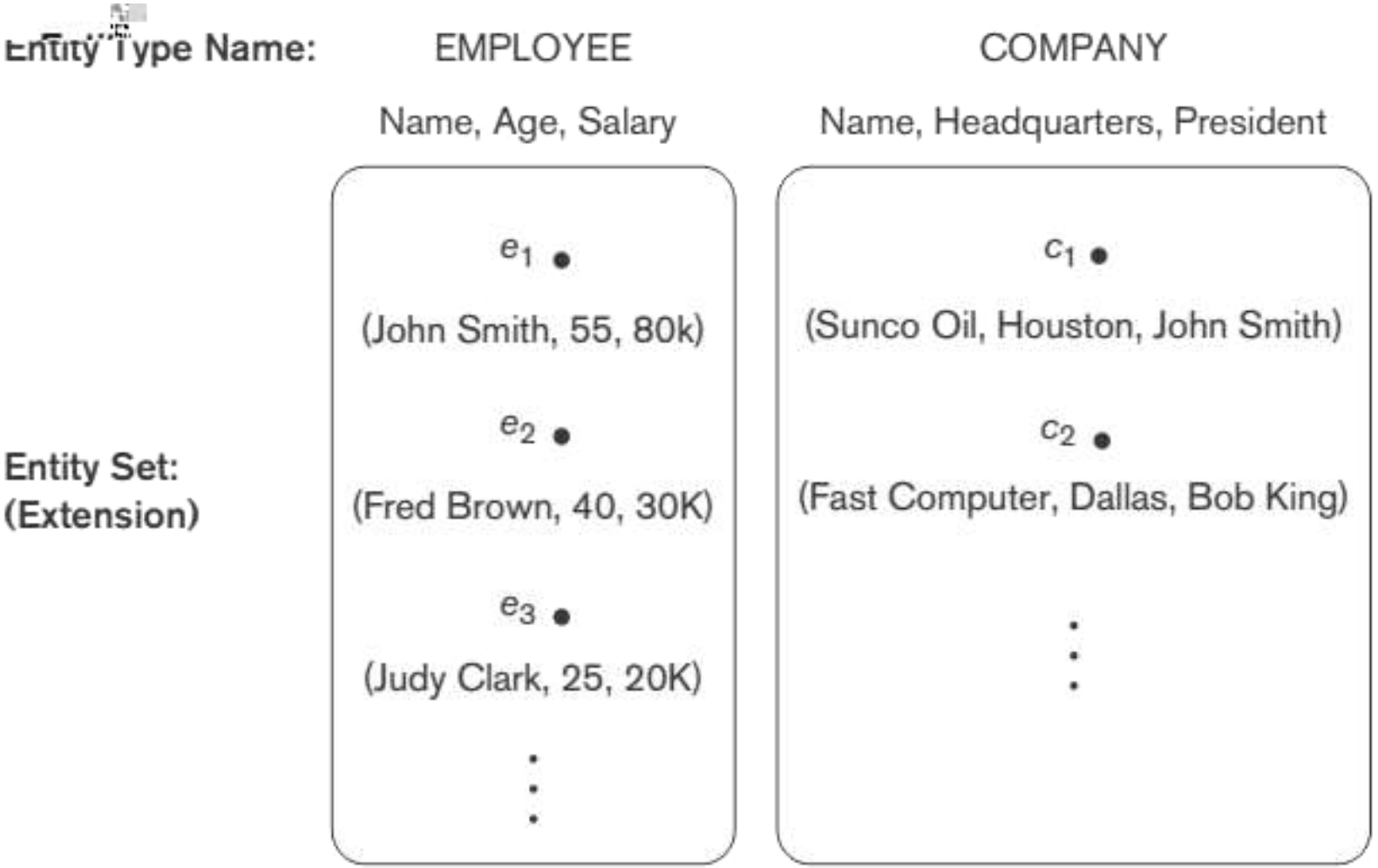
Entities and Attributes



Two entities, EMPLOYEE e1, and COMPANY c1, and their attributes.

Entity Types

- Collection (or set) of entities that have the same attributes

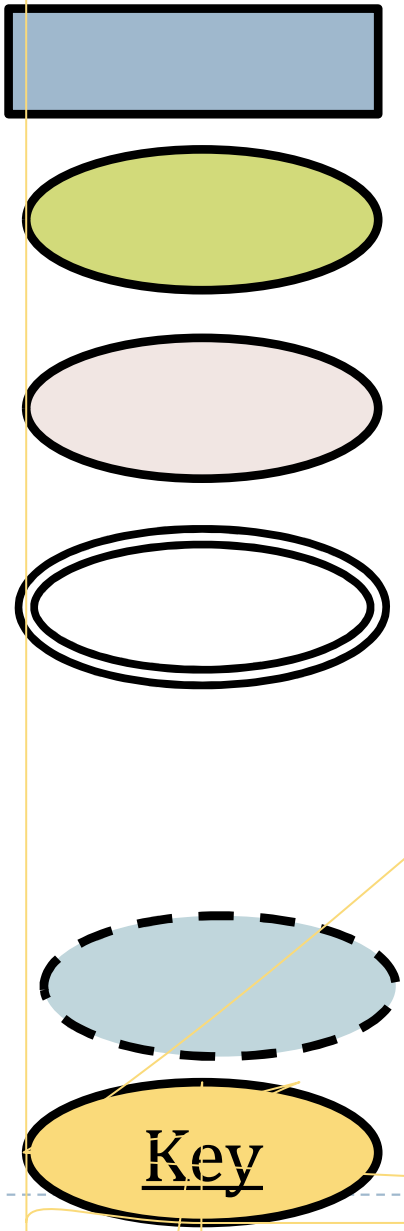


Keys

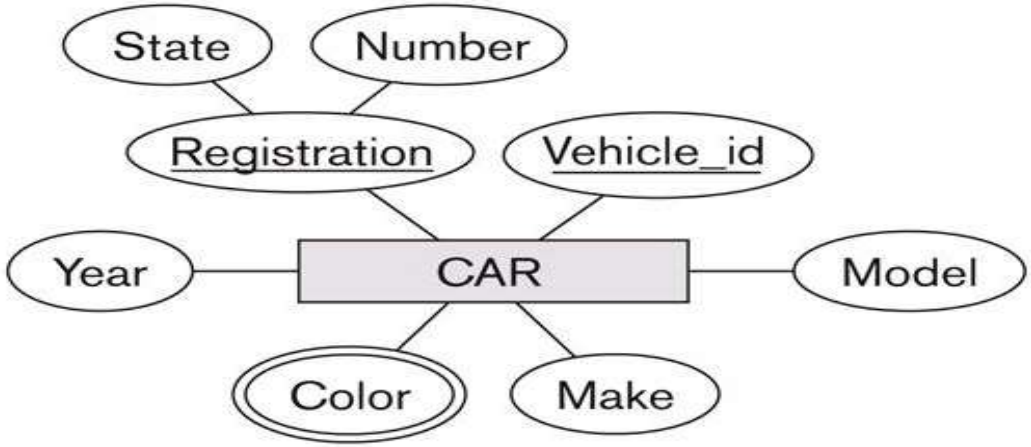
- **Key or uniqueness constraint**
  - Attributes whose values are **distinct** for each individual entity in entity set
  - Uniqueness property must hold for every entity set of the entity type
  - Ex: SSN of EMPLOYEE
- An entity type may have **more than one key**.
  - Ex: the STUDENT entity type may have two keys (in university context):
    - Citizen ID and
    - Student ID

Notations of Entity type, Attributes, Key

- Entity type
- Simple attribute
- Composite attribute
- Multi-valued attribute
- Complex attribute
- Derived attribute
- Key



Entity Type CAR with two keys and a corresponding Entity Set



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})

Identify Entity Types and Attributes

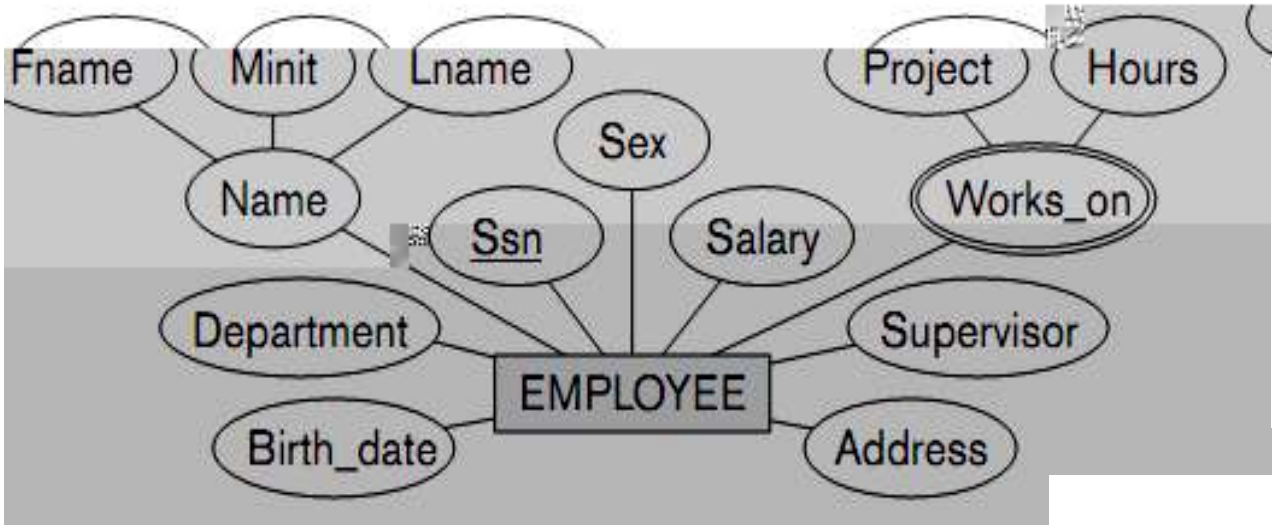
The COMPANY database:

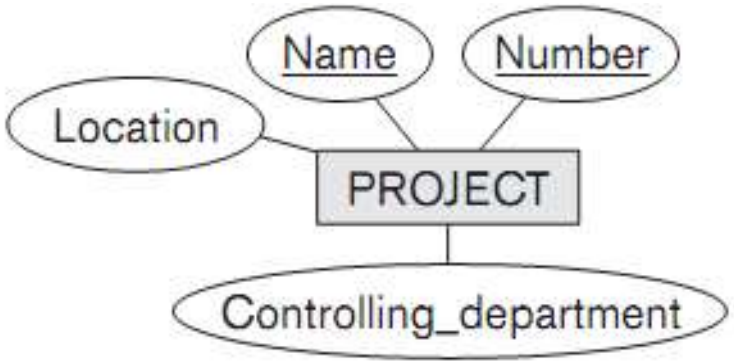
- The company is organized into **DEPARTMENTs**. Each department has *sl gpsc l k c sl gpsc l sk cp*, and a particular *ck n jmvcc u fmk l e cq* the department. We keep track of the *qr prb rc* when that employee began managing the department. A department may have *qct c p j jma rgml q*.
- A *bcn prk cl r aml rpmjq* a number of **PROJECTs**, each of which has *sl gpsc l k c sl gpsc l sk cp* and *qgj e jc jma rgml* .

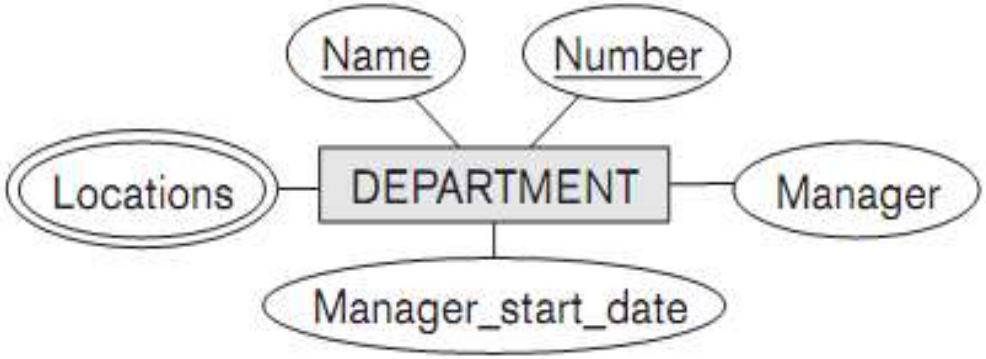
Identify Entity Types, Attributes

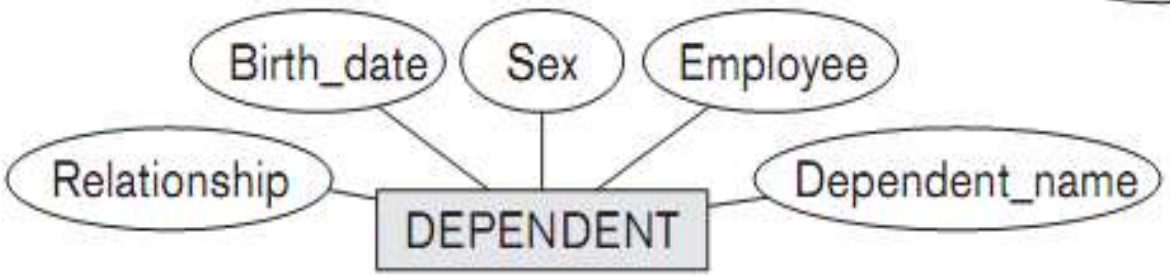
- We store **EMPLOYEE**'s *l k c mag j casp gwl sk cp bbpcqq q j pw qcv* and *gprf b rc*. An employee is *qqgl cb rmml c bcn prk cl r*; but may *u mpi ml qct cp jn pmhcarq*, which are not necessarily controlled by the same department. We keep track of the current *l sk cp ml fmspq ncpu cci* that an employee works on each project. We also keep track of the *bgpcar qsncpt gqmp* of each employee.
- We want to keep track of the **DEPENDENTs** of each employee, *gj ajsbgje dgpqrl k c qcv gprf b rc* and *pcj rgml qf gn* to the employee.

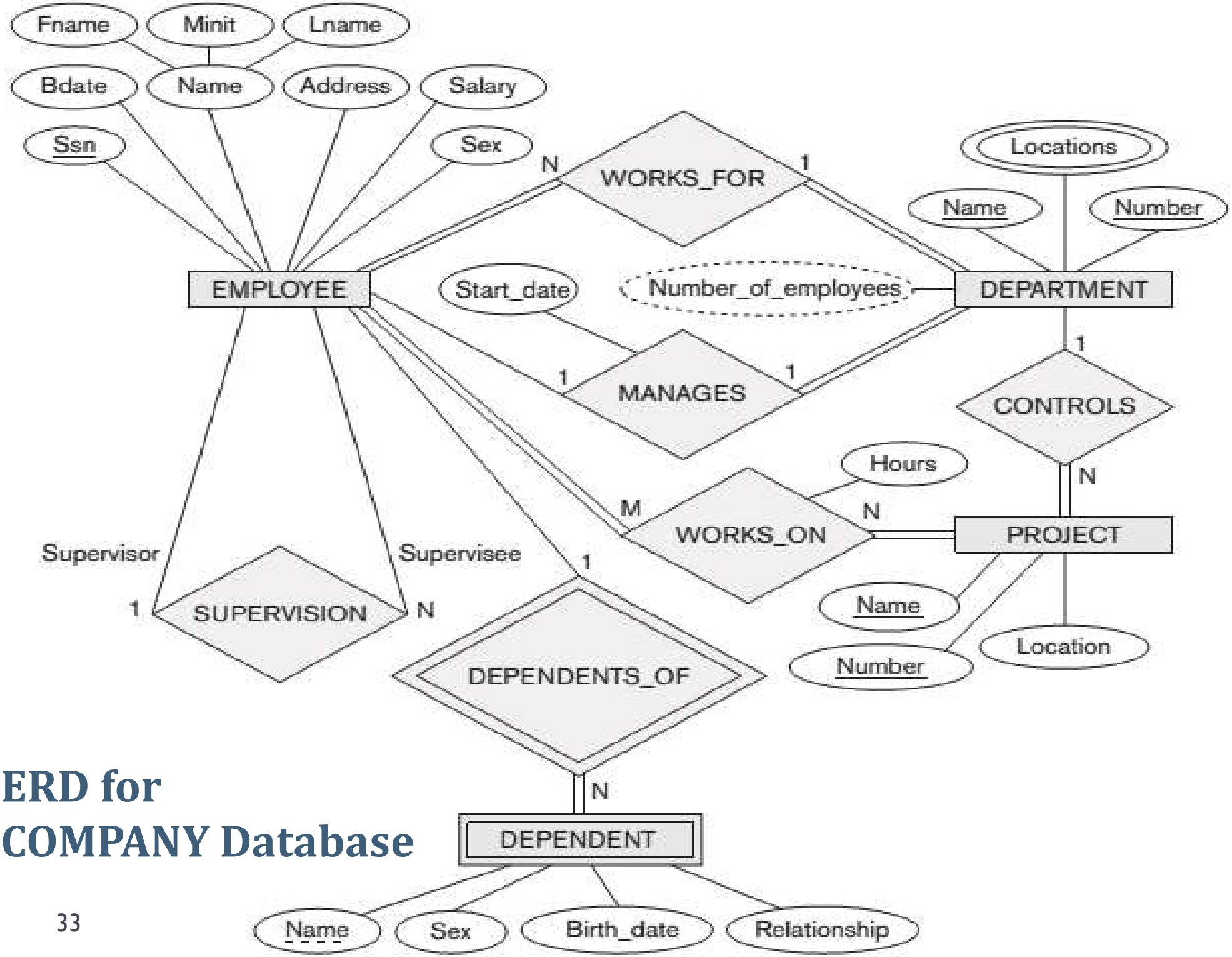
Initial Conceptual Design of COMPANY Database











ERD for COMPANY Database

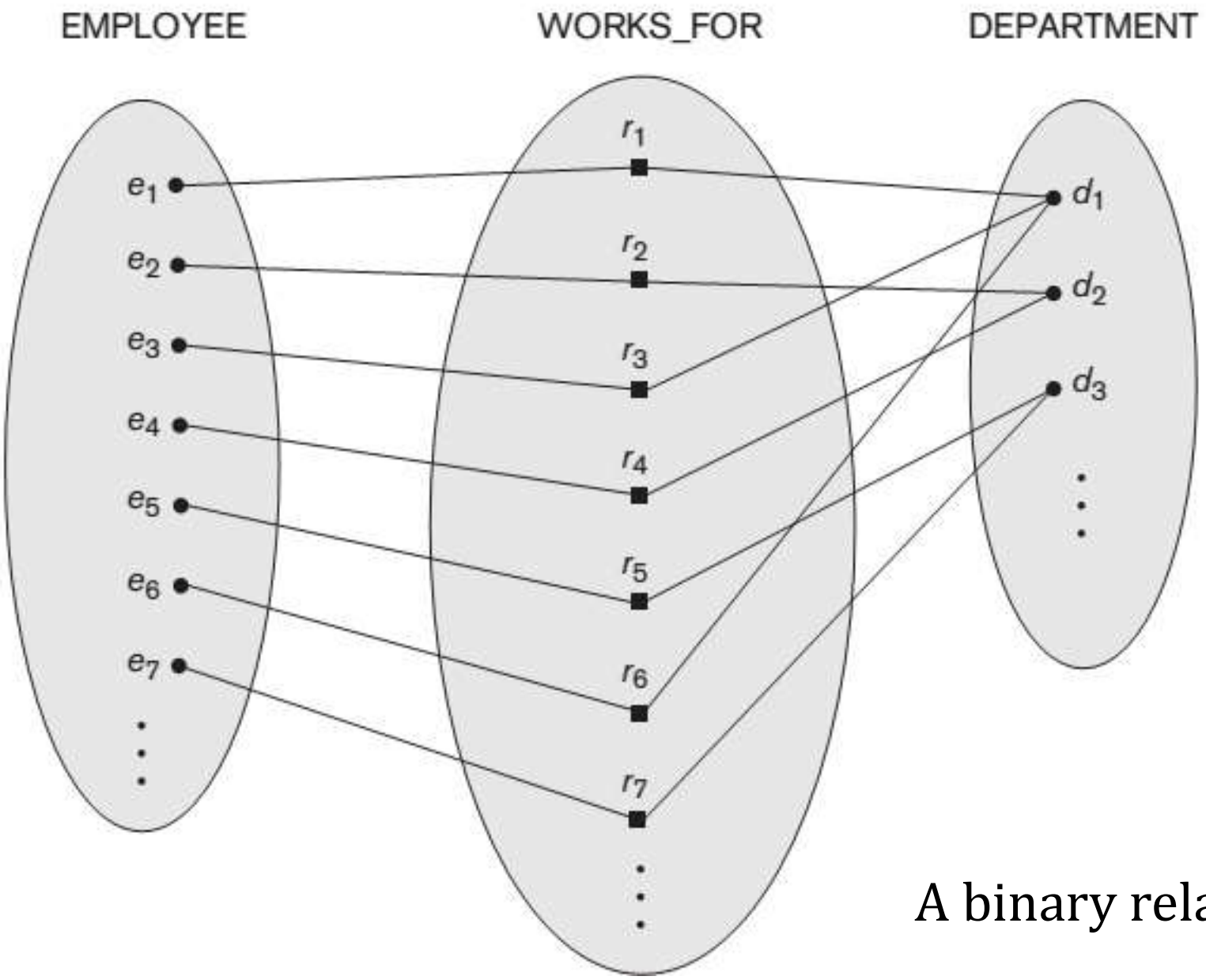
Relationships and Relationship Types

- **Relationship type** among l entity types C C C_l
  - Defines a set of associations among entities from these entity types
  - Ex: Relationship type WORKS_FOR between EMPLOYEEs and DEPARTMENTs
- **Relationship instances p_g**
  - Each p_g associates l individual entities (c c c_l). Each entity c_h in p_g is a member of entity set C_h
  - Ex: EMPLOYEE John Smith u m p i m l the PROJECT ProductX

Relationships and Relationship Types

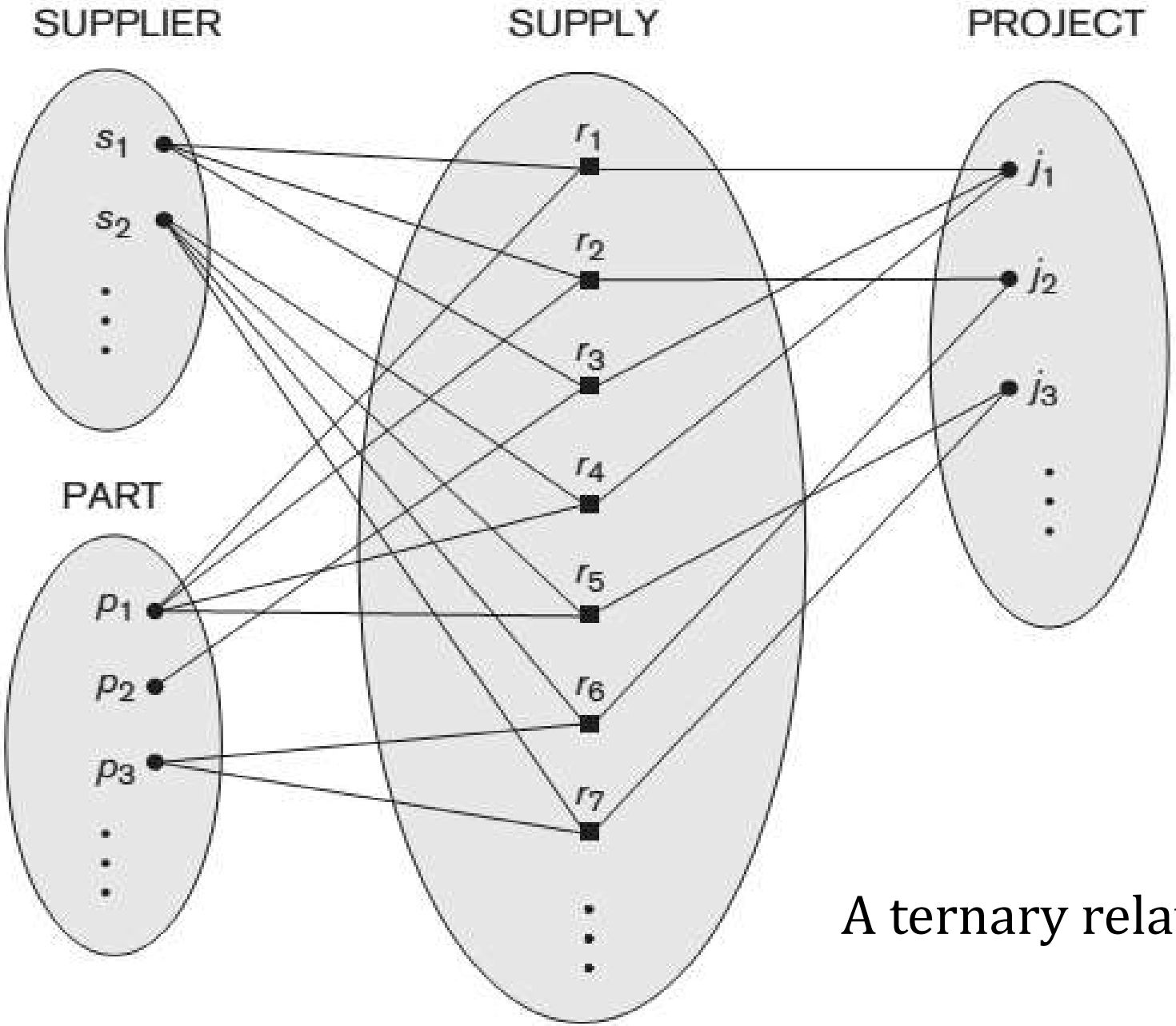
- **Degree** of a relationship type
  - Number of participating entity types
  - Binary (degree 2), ternary (degree 3), and n-ary (degree n)
- More than one relationship type can exist with the same participating entity types.
  - EMPLOYEE – *mpi .dmp* – DEPARTMENT
  - EMPLOYEE - *l e c* – DEPARTMENT

Example relationship instances



A binary relationship

Example relationship instances

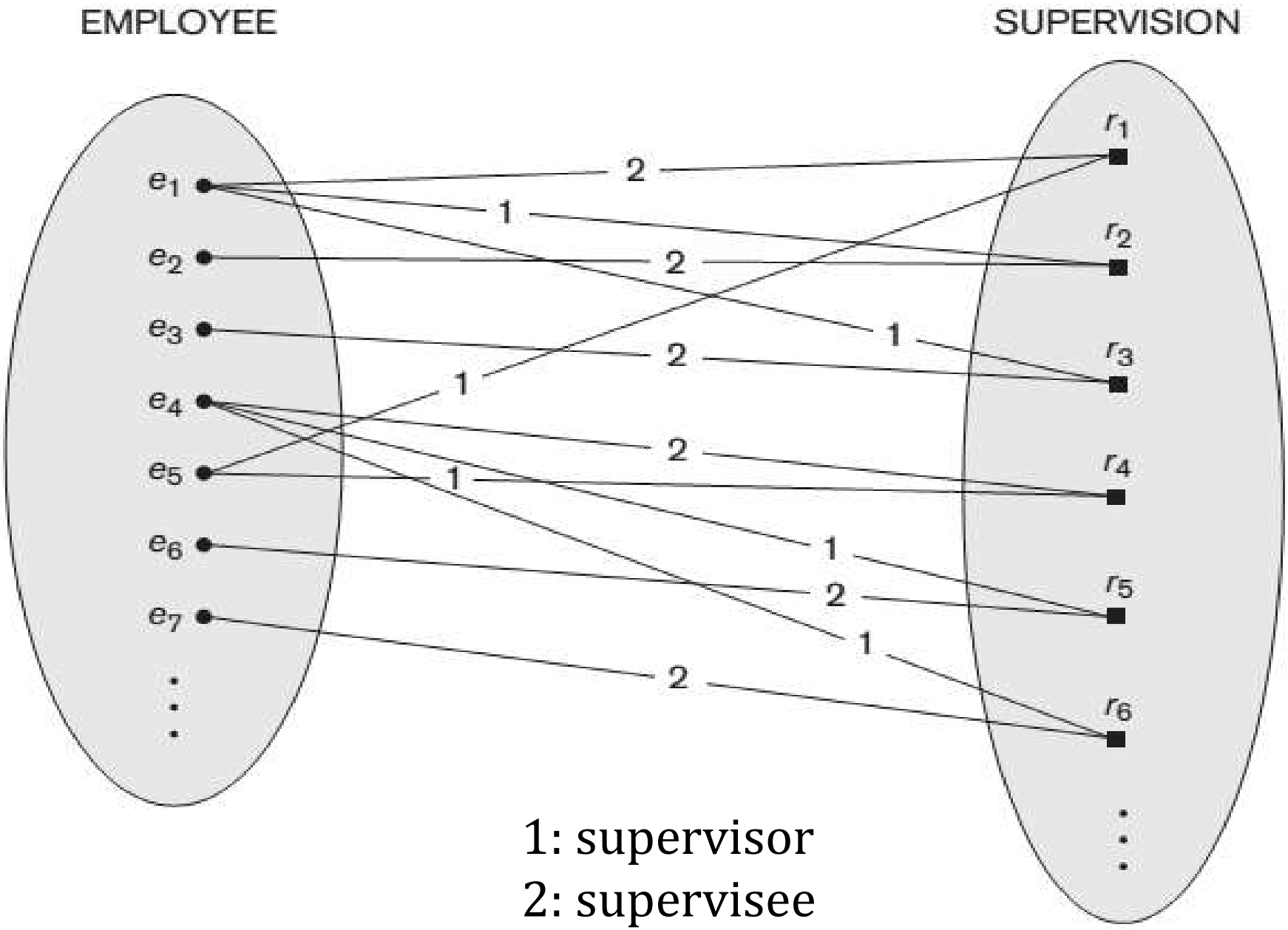


A ternary relationship

Relationships and Relationship Types

- **Recursive relationships**
  - Same entity type participates more than once in a relationship type in different roles
  - Must specify *prmjc* that a participating entity plays in each relationship instance
  - Ex: SUPERVISION relationships between EMPLOYEE (in role of supervisor or boss) and (another) EMPLOYEE (in role of subordinate or worker)

A Recursive Relationship SUPERVISION

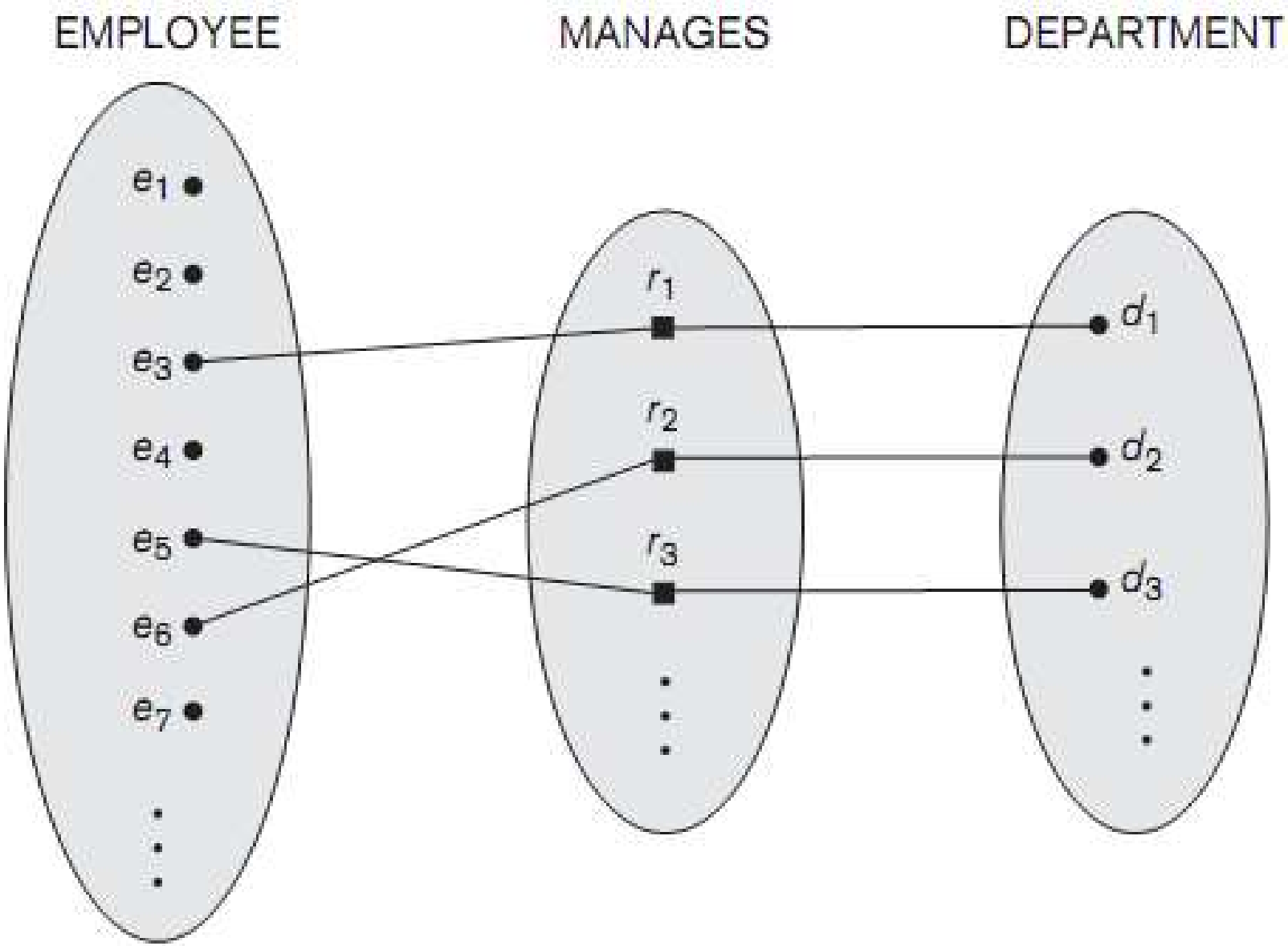


1: supervisor
2: supervisee

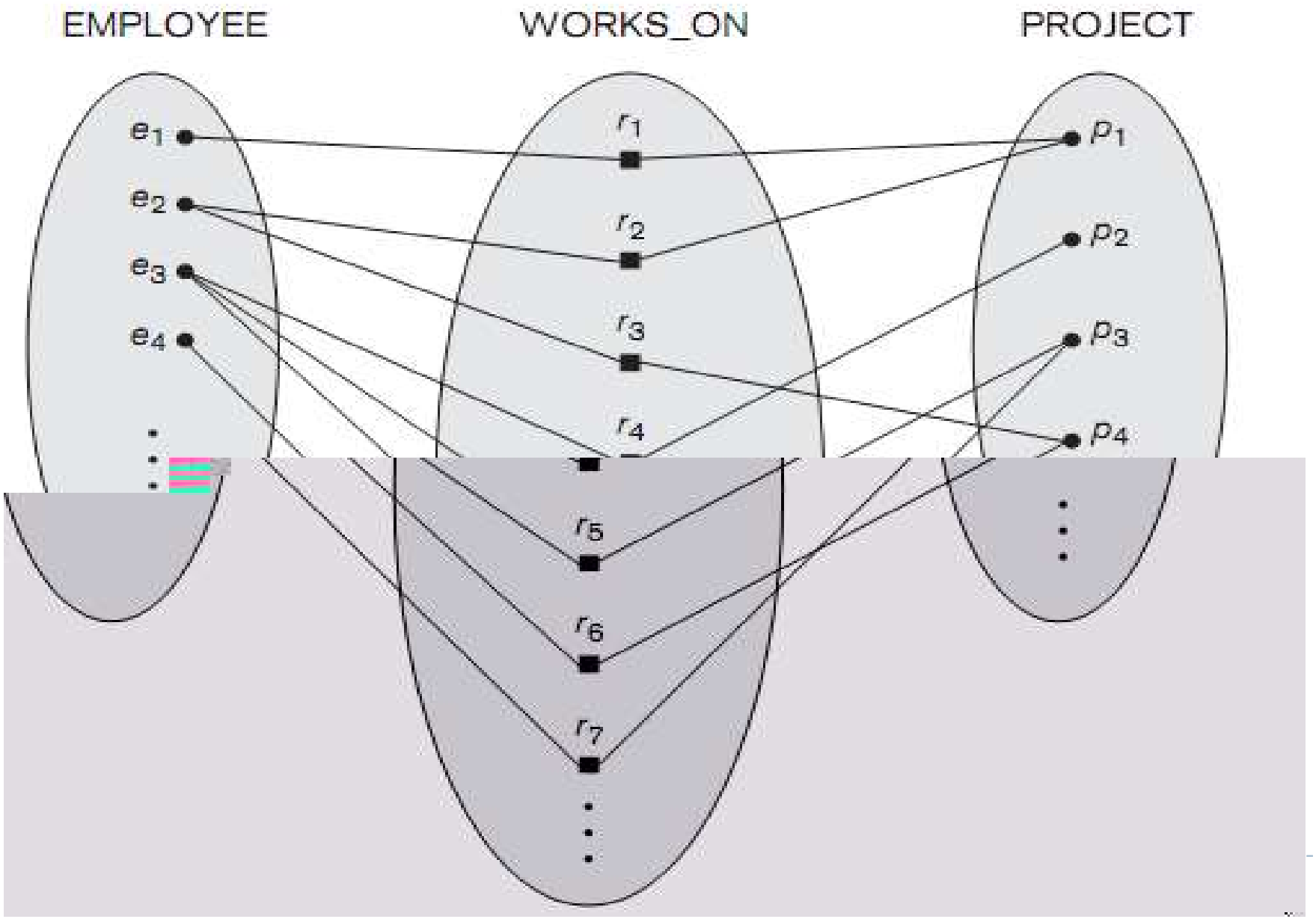
Constraints on Binary Relationship Type

- **Structural constraints**: one way to express semantics of relationship: *a pbdj jgwp rgm* and *n prgaggn rgml amlrp dj r*
- *pbdj jgwp rgm*, specifies maximum number of relationship instances that 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)

One-to-one (1:1) RELATIONSHIP



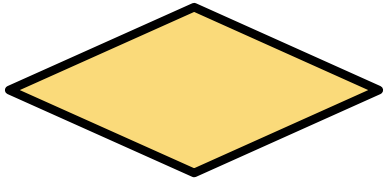
Many-to-many (M:N) RELATIONSHIP



Constraints on Binary Relationship Type

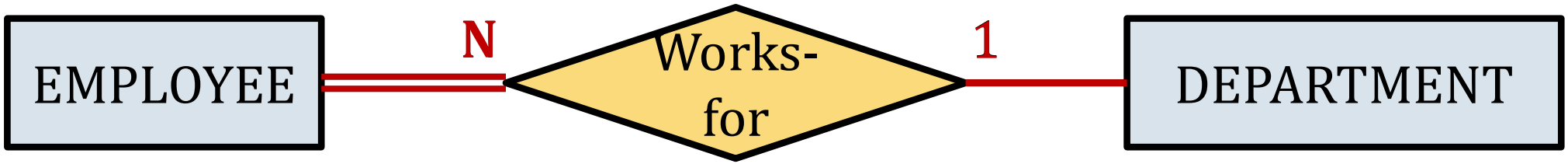
- Participation constraint (or membership class): specifies whether existence of entity depends on its being related to another entity
  - *l b rmpw* (total participation) - every instance of a participating entity type must participate in the relationship. (double line)
  - *Mnrgml j* (partial participation) - not every instance of a participating entity type must participate in the relationship. (single line)

Notations of Relationship type

- Relationship type 

Cardinality ratio

An EMPLOYEE works for **one** DEPARTMENT.
A DEPARTMENT has **many** EMPLOYEEs.

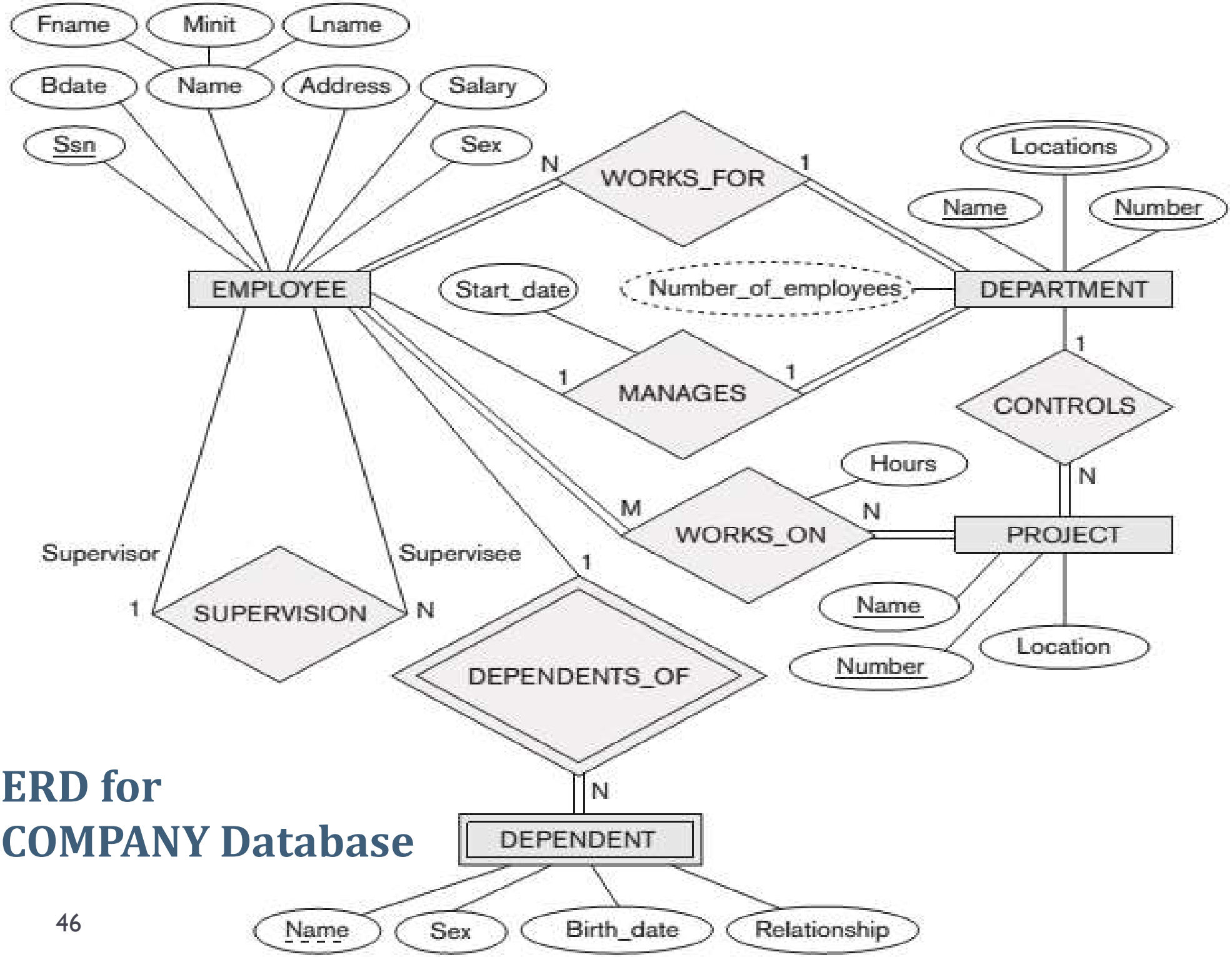


An EMPLOYEE **must** works for a DEPARTMENT.
A DEPARTMENT **may** have **no** EMPLOYEE.

Participation constraint

Attributes of Relationship Types

- A relationship type can have attributes.
  - HoursPerWeek of WORKS_ON
- 1:1 relationship type: relationship attributes can be migrated to any participating entity type.
- 1:N relationship type: relationship attributes can be migrated only to entity type on N-side of relationship.
- M:N relationship types: relationship attributes cannot be migrated to any entity type.



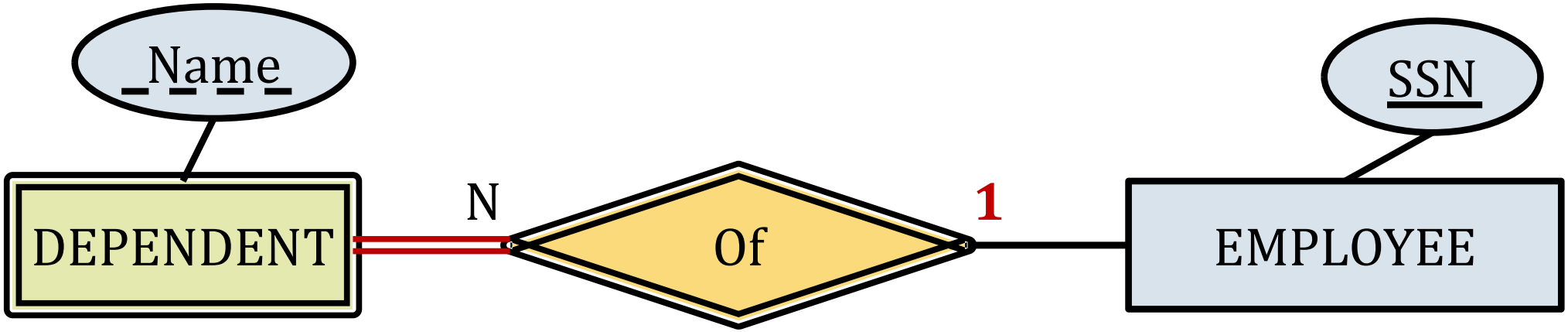
ERD for COMPANY Database

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
- 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

Notations of Relationship type

- Weak entity type
- Identifying relationship type
- Partial key



Identifying/Parent/
Owner entity type

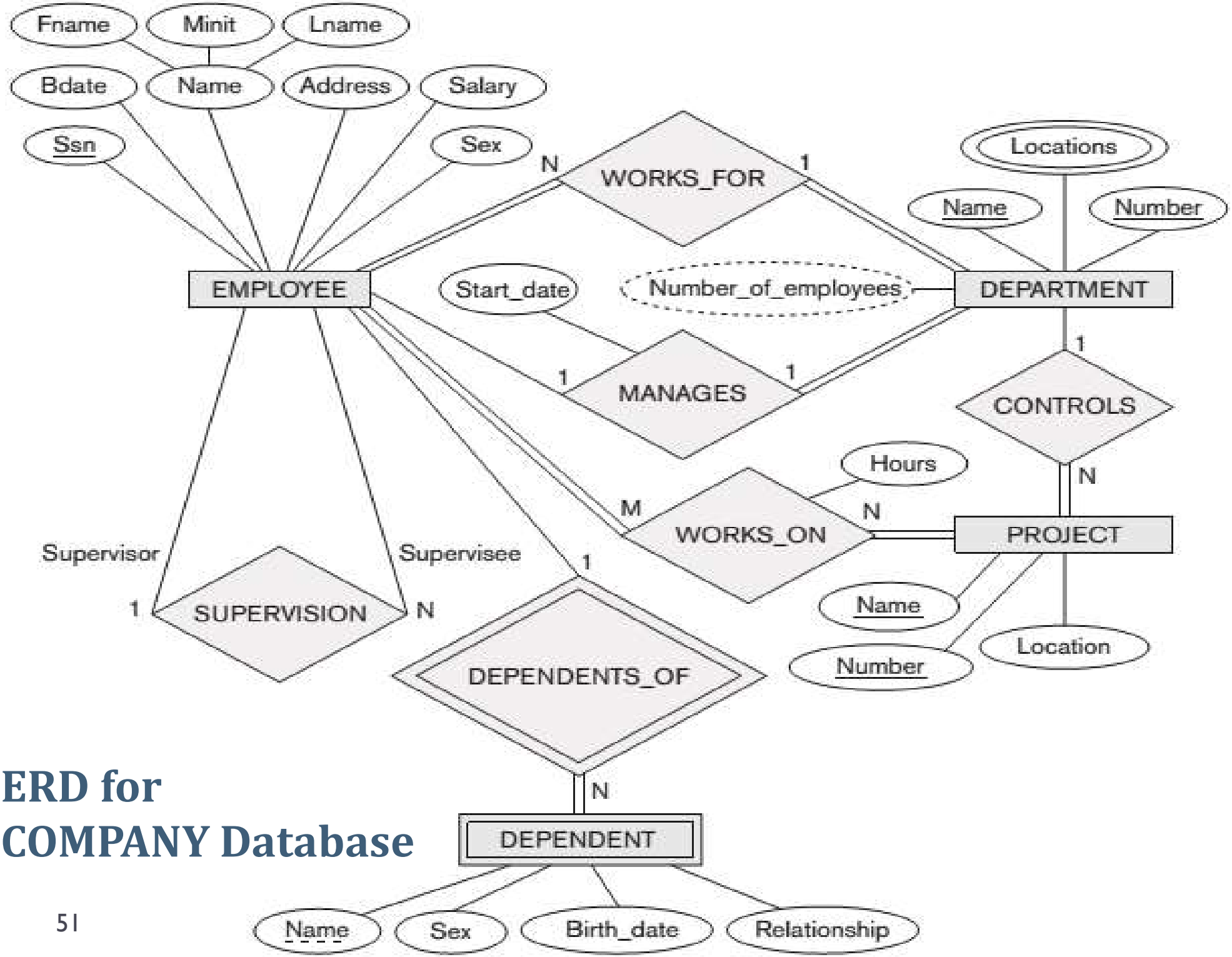
Identify Entity Types, Attributes, Relationships

The COMPANY database:

- The company is organized into **DEPARTMENTs**. Each department has *sl gpsc l k c sl gpsc l sk cp*, and a particular employee who *k l e cq* the department. We keep track of the *r prb rc* when that employee began managing the department. A department may have *qct c p j jma rgml q*.
- A department controls a number of **PROJECTs**, each of which has *sl gpsc l k c sl gpsc l sk cp* and *qgj e jc jma rgml* .

Identify Entity Types, Attributes, Relationships

- We store **EMPLOYEE**'s *l k c mag j casp gwl sk cp bbpcqq q j pw qcv* and *gprf b rc*. An employee is *qqgg l cb* to one department, but may *u mpi ml* several projects, which are not necessarily controlled by the same department. We keep track of the current *l sk cp ml fmsp ncpu cci* that an employee works on each project. We also keep track of the *b gpcarqsncpt gqmp* of each employee.
- We want to keep track of the **DEPENDENTs** of each employee, *gj ajsbgje dgpqrl k c qcv gprf b rc* and *pcj rgml qf gn* to the employee.



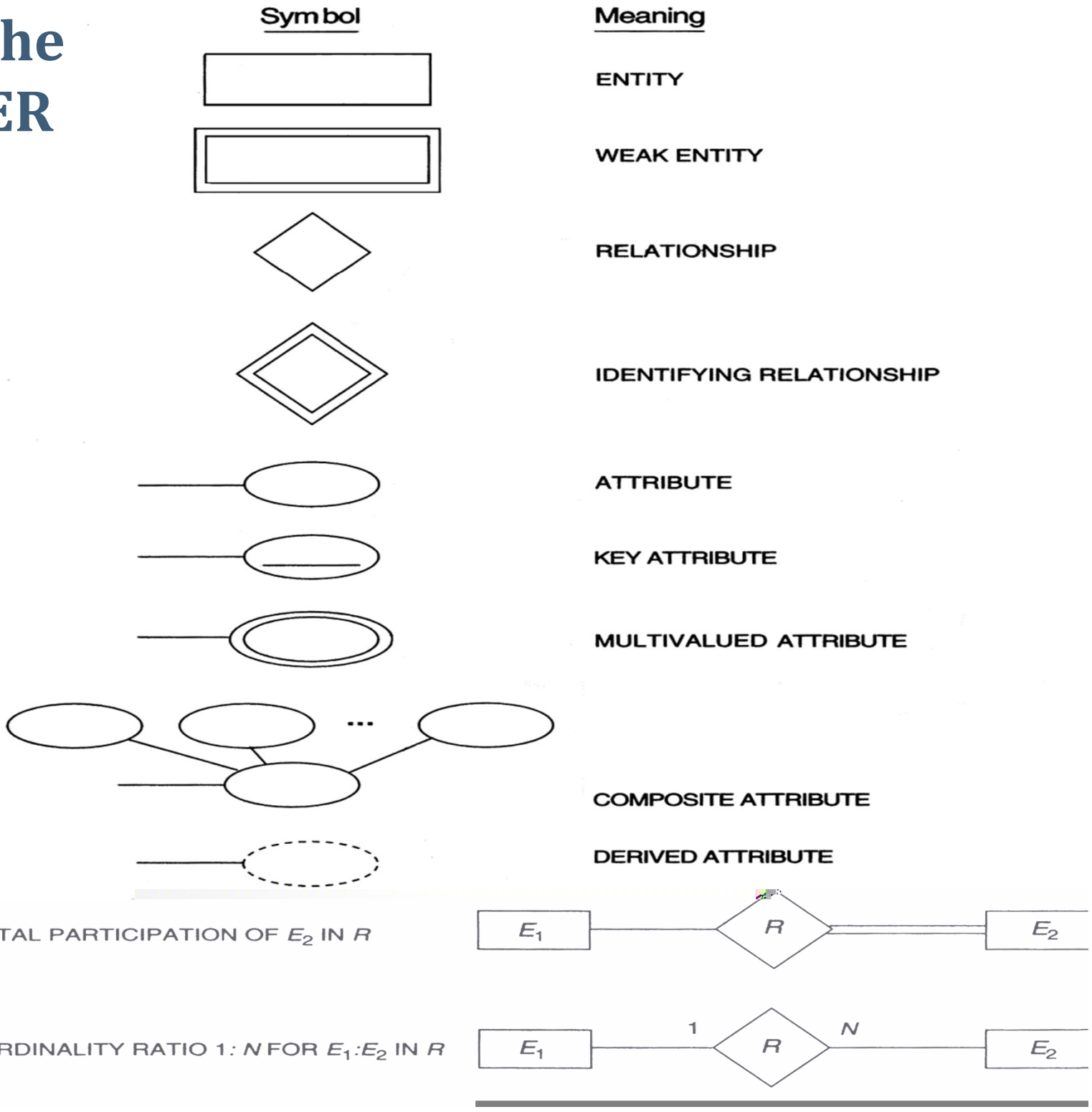
ERD for COMPANY Database

Contents

ER Diagram and Naming Conventions

- An ER model can be expressed in the form of the ER diagram.
- Proper Naming of Schema Constructs:
  - Choose names that convey meanings attached to different constructs in schema
  - *ms l* give rise to *cl rgw rwnc* names
  - *cp* indicate names of *pc j rgml fgn rwnc*
  - Choose binary relationship names to make ER diagram readable from left to right and from top to bottom

Summary of the Notation for ER Diagrams



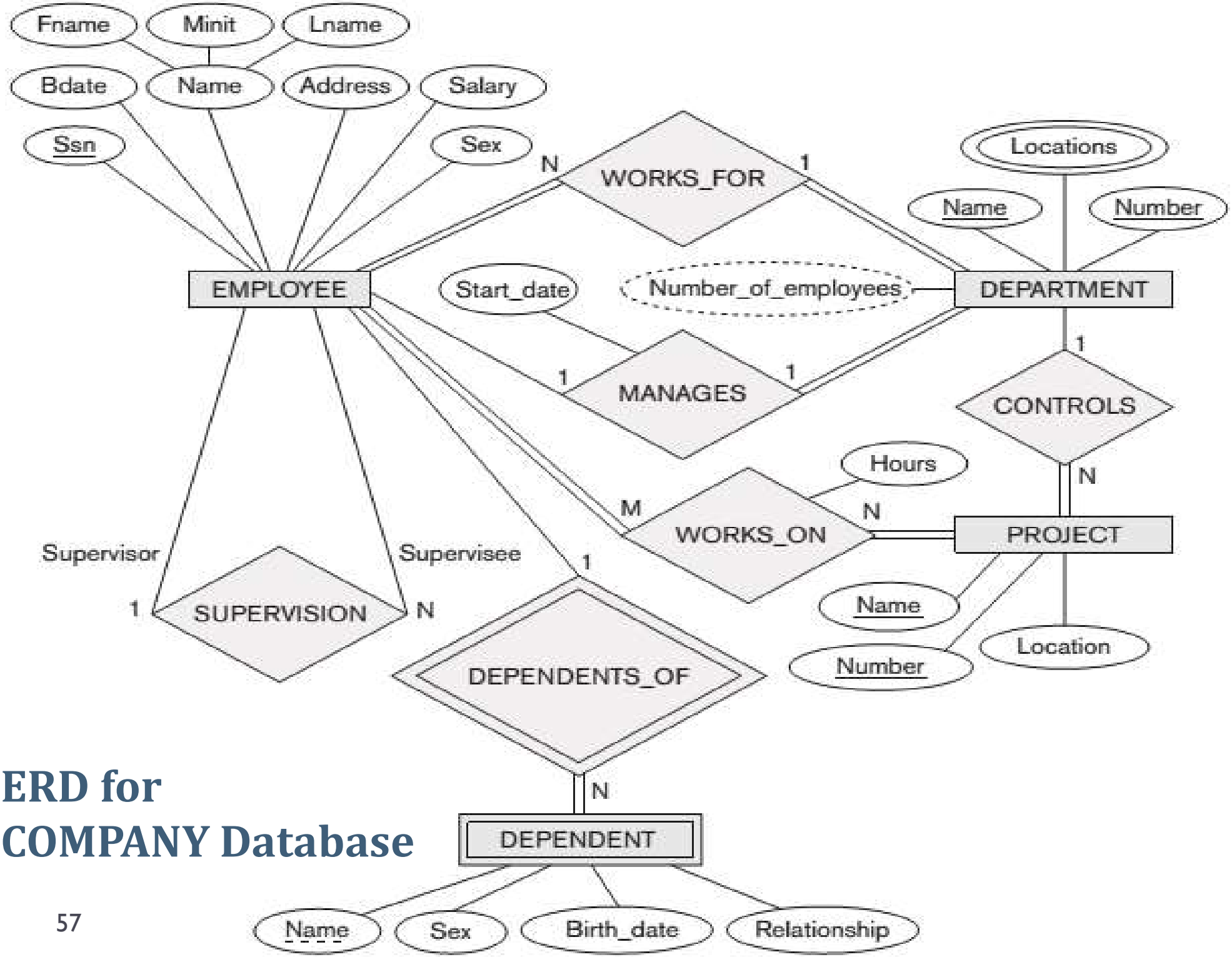
Draw Entity-Relationship Diagram

The COMPANY database:

- The company is organized into **DEPARTMENTs**. Each department has *sl gpsc l k c sl gpsc l sk cp*, and a particular employee who *k l e cq* the department. We keep track of the *r prb rc* when that employee began managing the department. A department may have *qct c p j jma rgml q*.
- A department controls a number of **PROJECTs**, each of which has *sl gpsc l k c sl gpsc l sk cp* and *qgj e jc jma rgml* .

Draw Entity-Relationship Diagram

- We store **EMPLOYEE**'s *l k c mag j casp gwl sk cp bbpcqq q j pw qcv* and *gprf b rc*. An employee is *qqgg l cb* to one department, but may *u mpi ml* several projects, which are not necessarily controlled by the same department. We keep track of the current *l sk cp ml fmsp ncpu cci* that an employee works on each project. We also keep track of the *b gpcarqsncpt gqmp* of each employee.
- We want to keep track of the **DEPENDENTs** of each employee, *gj ajsbgje dgpqrl k c qcv gprf b rc* and *pc j rgml qf gn* to the employee.



ERD for COMPANY Database

Case study: Draw ERD

GROUP A	GROUP B
A system for course registration of HCMUT	An order management system for a shop

Case study: Draw ERD

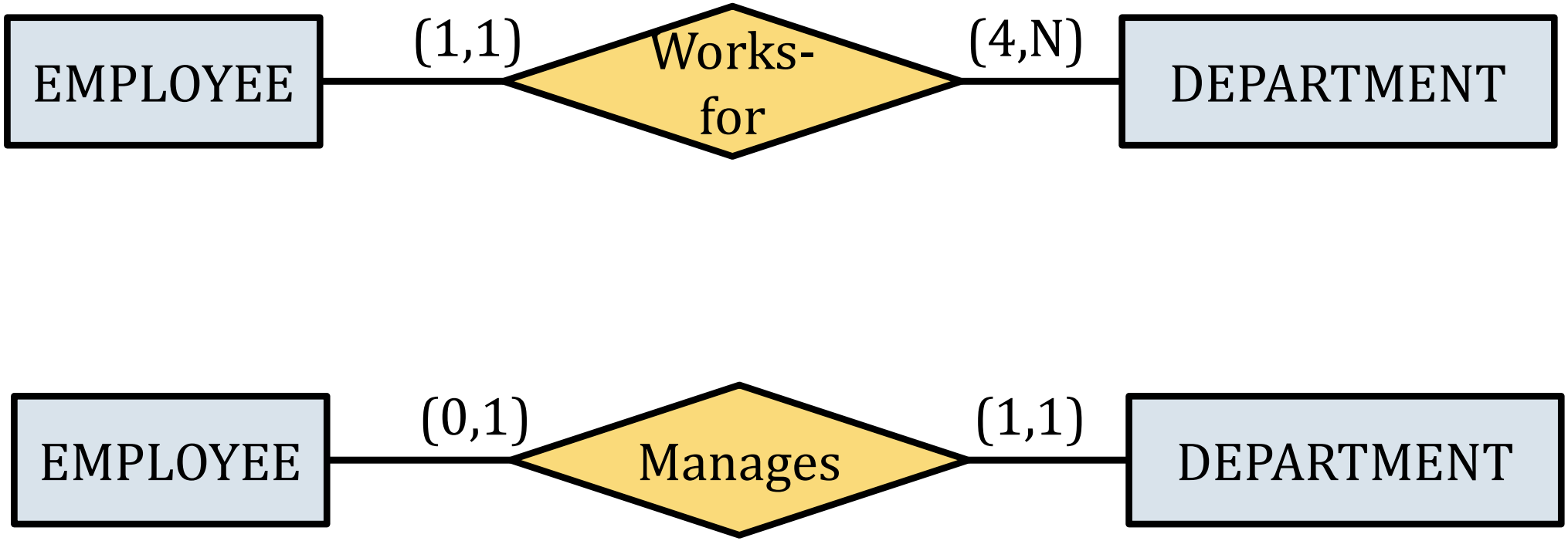
GROUP A	GROUP B
A system for course registration of HCMUT	A system for a Library of a University

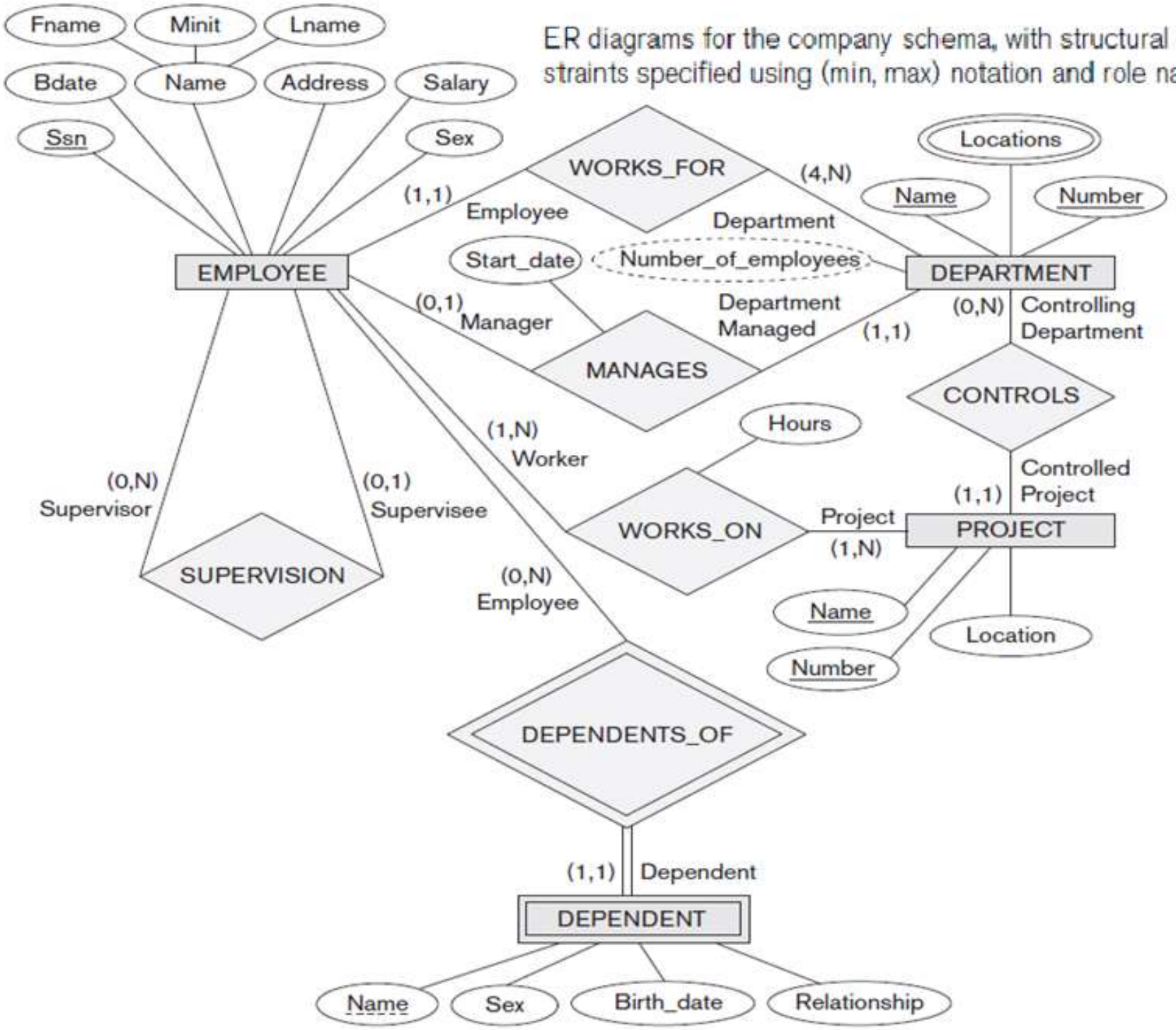
Contents

Alternative Diagrammatic Notations

- **(Min-max) notation for relationships**
  - Specify structural constraints on relationships
  - Replaces cardinality ratio (1:1, 1:N, M:N) and single/double line notation for participation constraints
  - Associate a pair of integer numbers (min, max) with each participation of an entity type E in a relationship type R, where $0 \leq \text{min} \leq \text{max}$ and $\text{max} \geq 1$

(min, max) notation

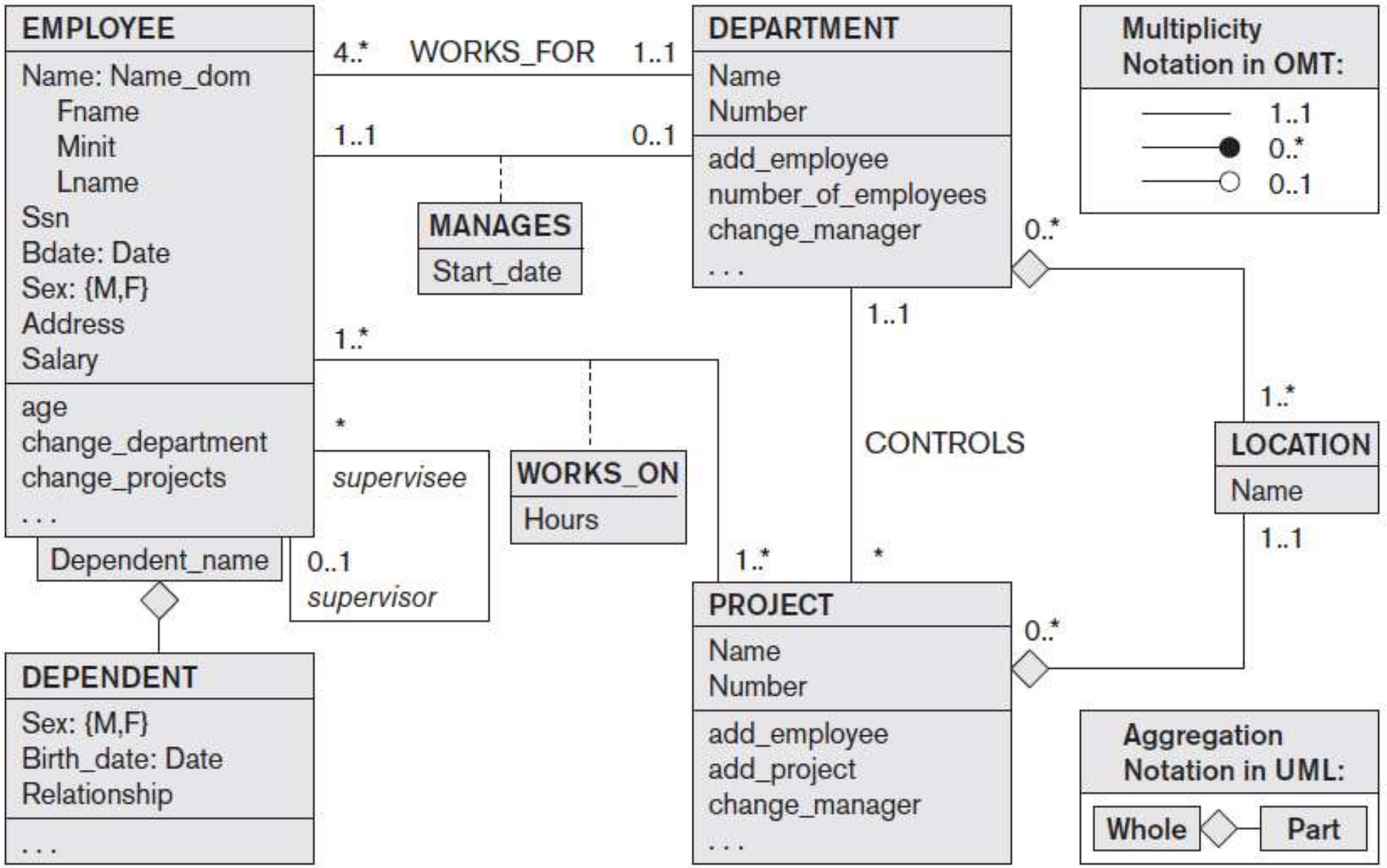



ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.
Fname
Minit
Lname
Name
Bdate
Address
Salary
Ssn
Sex
WORKS_FOR
(1,1)
Employee
(4,N)
Department
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
(0,1)
Manager
MANAGES
Department Managed
(1,1)
(0,N)
Controlling Department
CONTROLS
Hours
(1,N)
Worker
Controlled Project
(1,1)
(0,N)
Supervisor
(0,1)
Supervisee
WORKS_ON
Project
(1,N)
PROJECT
SUPERVISION
(0,N)
Employee
Name
Location
Number
DEPENDENTS_OF
(1,1)
Dependent
DEPENDENT
Name
Sex
Birth_date
Relationship

Alternative Diagrammatic Notations

- UML methodology
  - Used extensively in software design
  - Many types of diagrams for various software design purposes
- **UML class diagrams**
  - Entity in ER corresponds to an object in UML

The COMPANY conceptual schema in UML class diagram notation.



Alternative Diagrammatic Notations

- **UML class diagrams**
  - **Class** includes three sections:
    - Top section gives the class name
    - Middle section includes the attributes;
    - Last section includes operations that can be applied to individual objects
  - **Associations**: relationship types
  - **Relationship instances**: links

Alternative Diagrammatic Notations

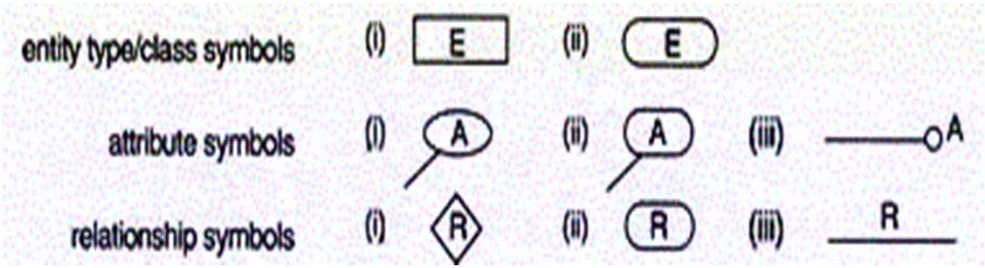
- **UML class diagrams**
  - Binary association
    - Represented as a line connecting participating classes
    - May optionally have a name
  - Link attribute
    - Placed in a box connected to the association's line by a dashed line

Alternative Diagrammatic Notations

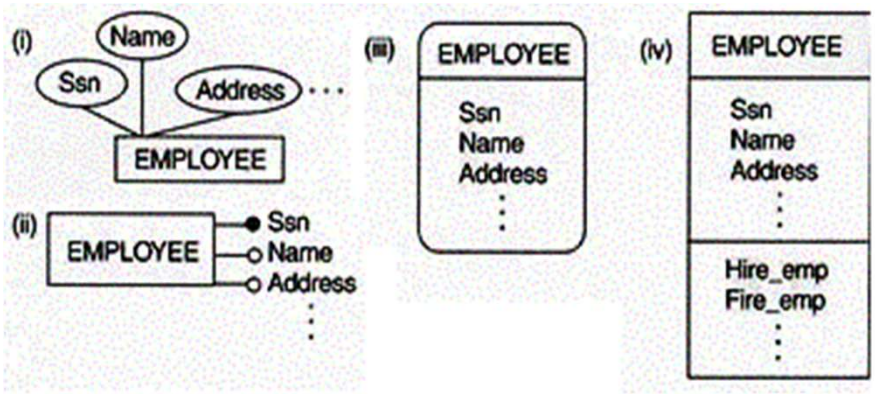
- **UML class diagrams**
  - **Multiplicities**: min..max, asterisk (*) indicates no maximum limit on participation
  - Types of relationships: **association** and **aggregation**
  - Distinguish between **unidirectional** and **bidirectional** associations
  - Model weak entities using **qualified association**

Alternative Diagrammatic Notations

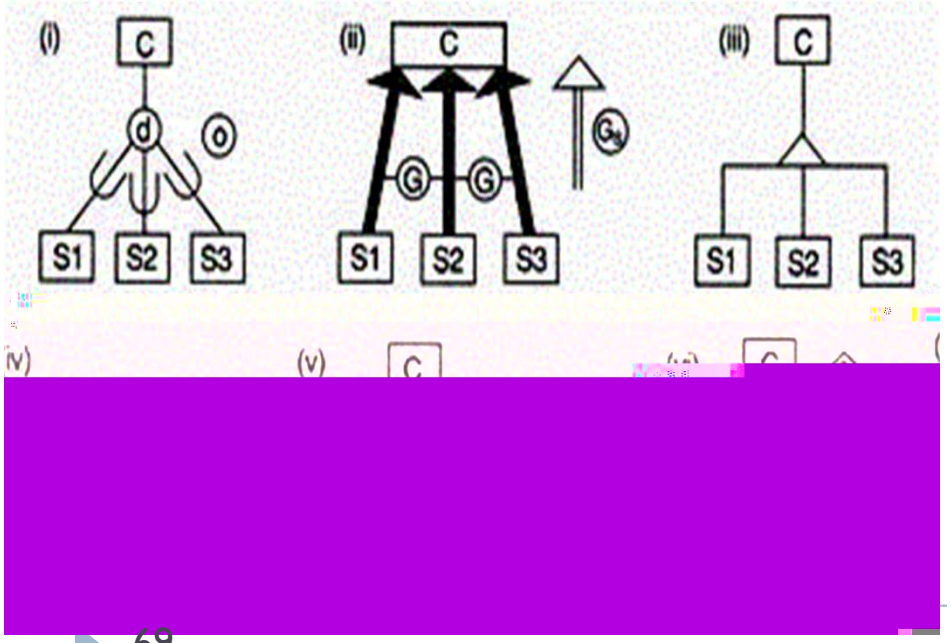
Symbols for entity type / class, attribute and relationship



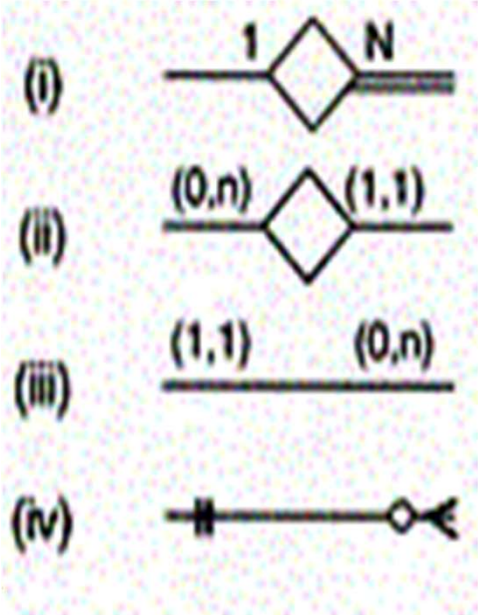
Displaying attributes



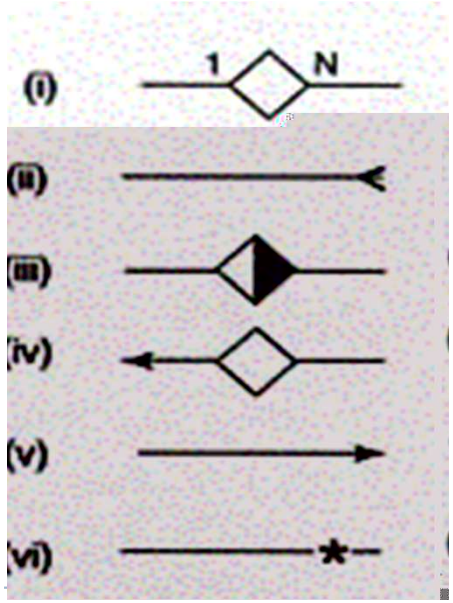
Notations for displaying specialization / generalization



Various (min, max) notations



Displaying cardinality ratios



Contents

Problems with ER Models

- Semantic constraints
- Connection traps

Semantic constraints

- Constraints that cannot be directly expressed in the ER diagram
- Must be expressed and enforced by the trigger mechanism, or application programs or in some other ways.
- Examples:
  - The age of an employee must be greater than 18 years old
  - The salary of a department manager must be higher than the other employees works for that department.
  - When increasing salary of employee, the increasing amount must not more than 20% of current salary.

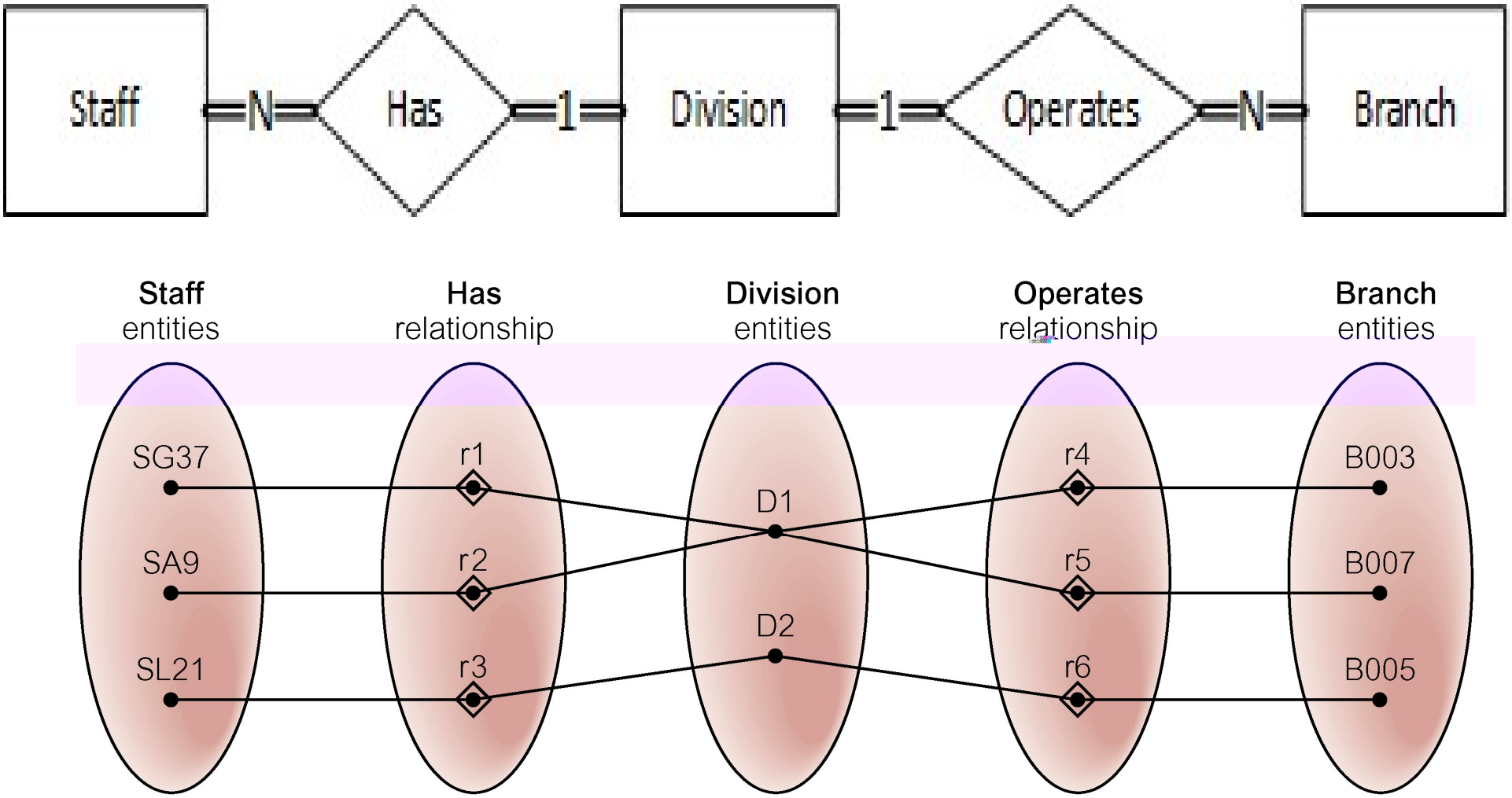
Connection traps

- Often due to a misinterpretation of the meaning of certain relationships
- Two main types of connection traps are called fan traps and chasm traps

Connection traps

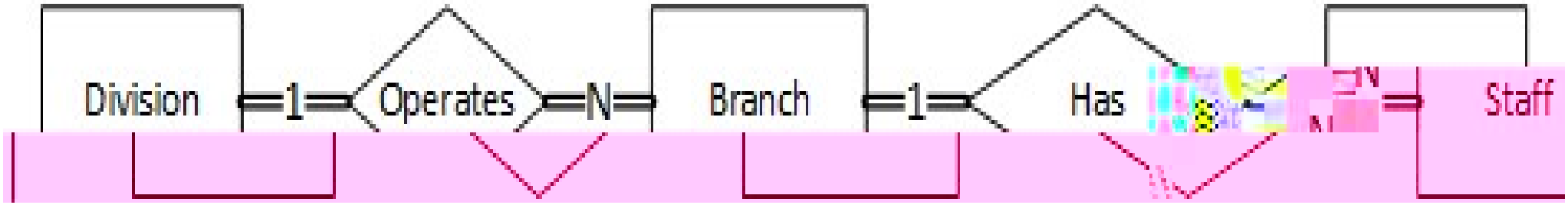
- **Fan Trap**
  - Where a model represents a relationship between entity types, but pathway between certain entity occurrences is ambiguous
  - Usually: two or more 1:N relationships fan out from the same entity
- **Chasm Trap**
  - Where a model suggests the existence of a relationship between entity types, but pathway does not exist between certain entity occurrences
  - Usually: optional participation

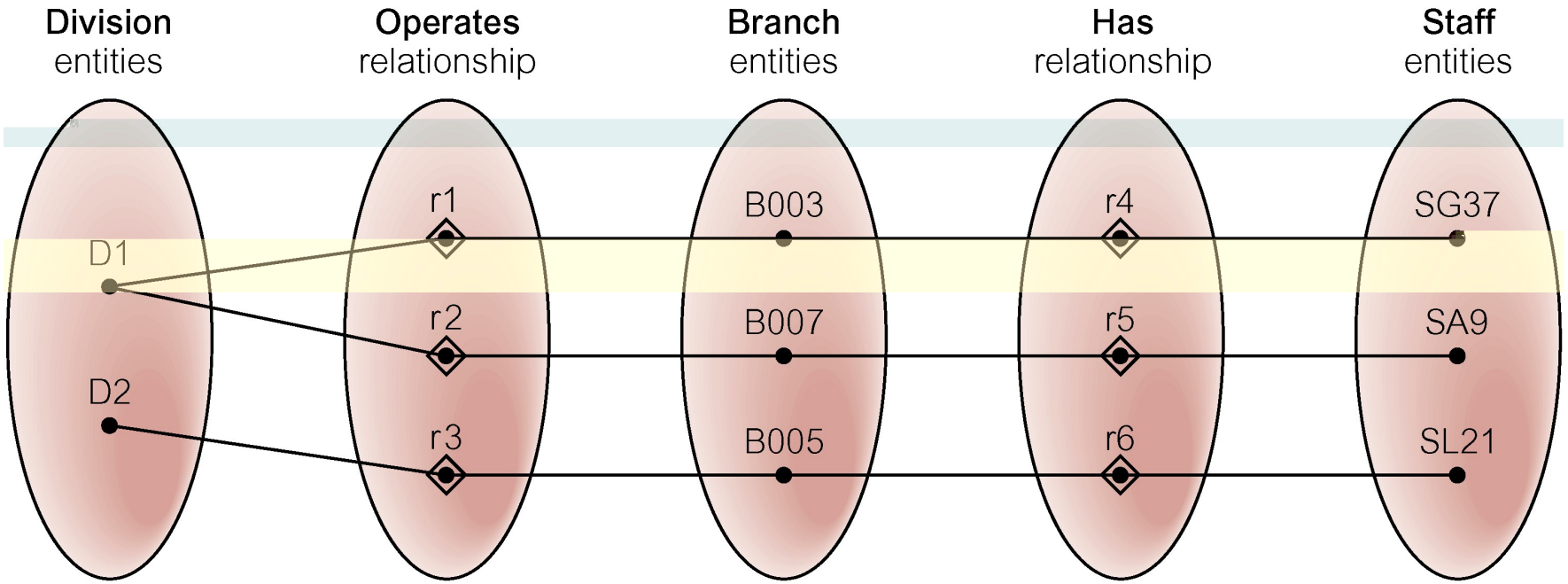
An Example of a Fan Trap



At which branch office does staff number SG37 work?

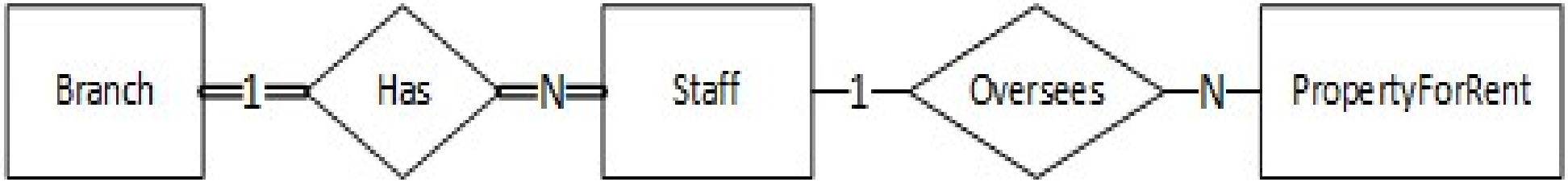
Restructuring ER model to remove Fan Trap

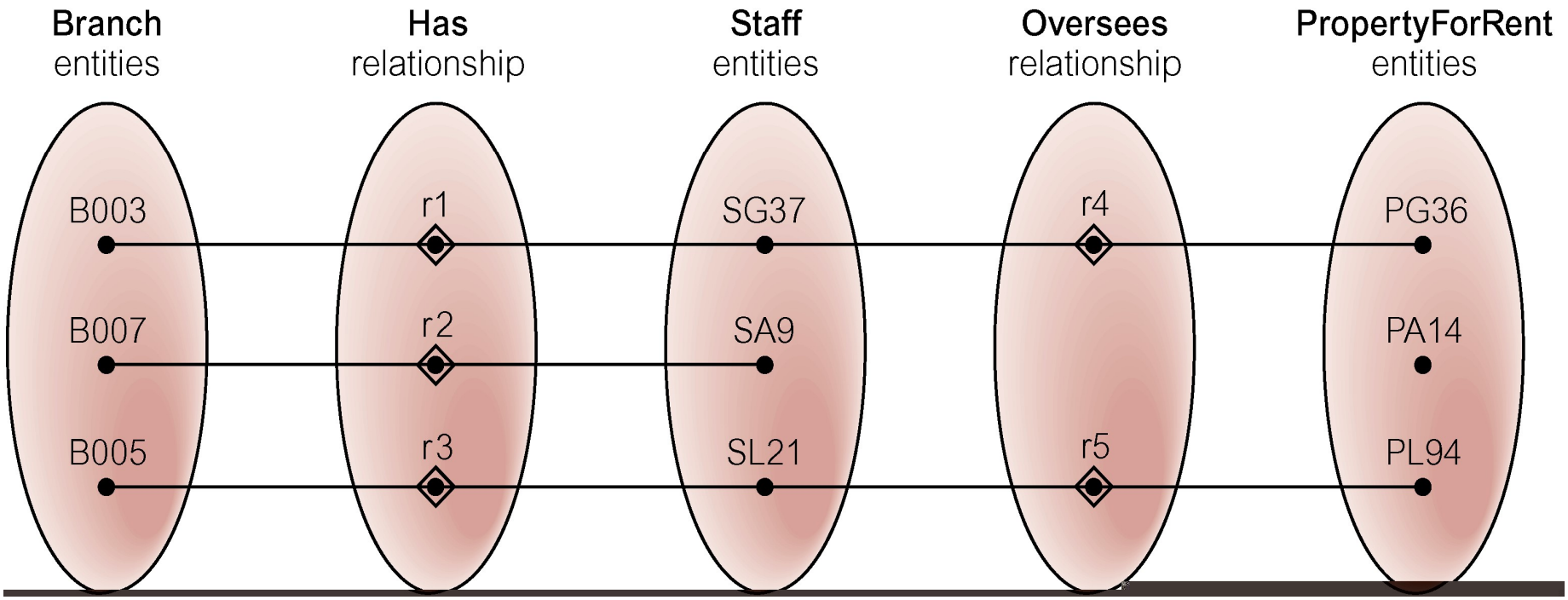




SG37 works at branch B003

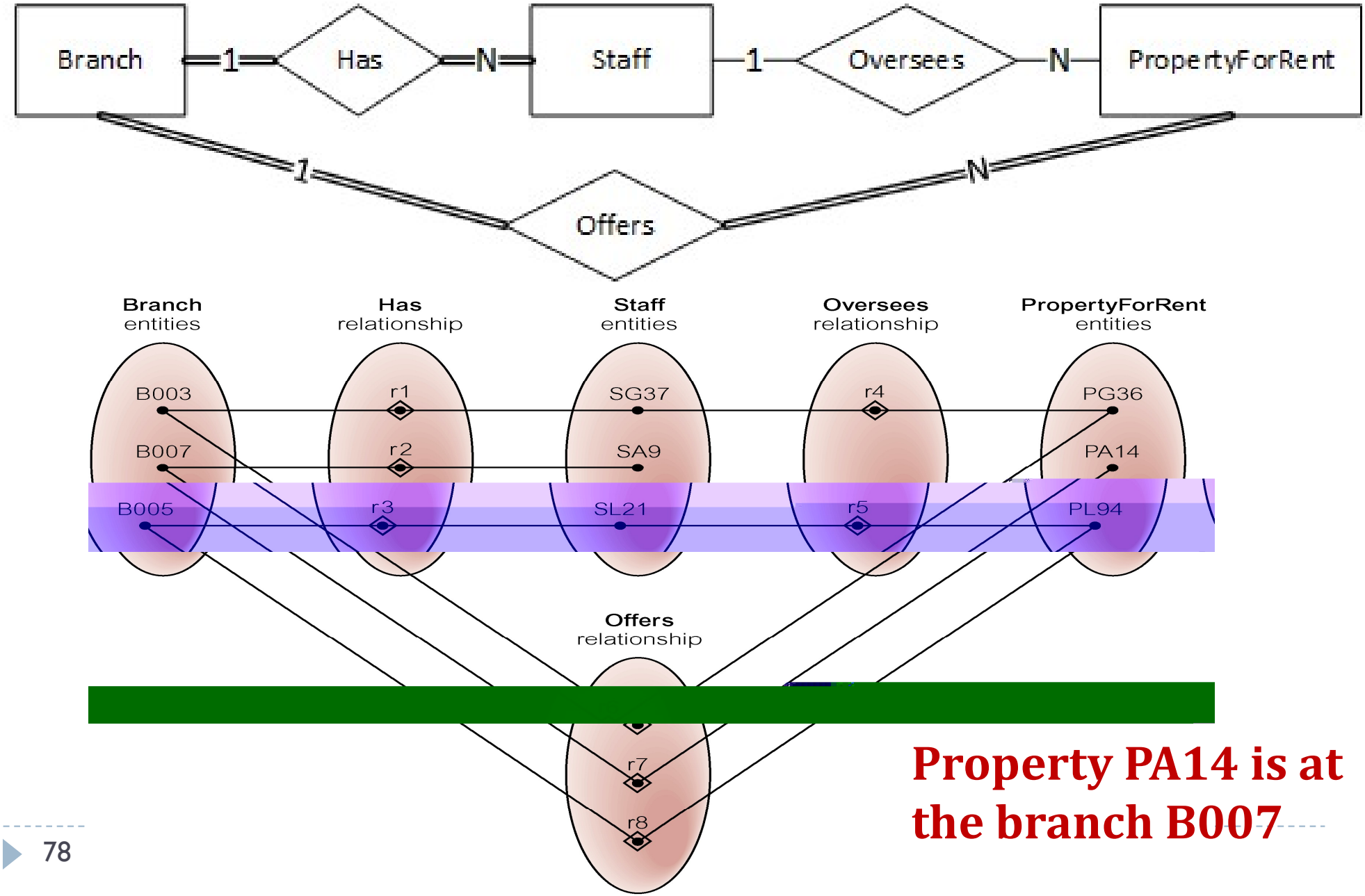
An Example of a Chasm Trap





At which branch office is property PA14 available?

ER Model restructured to remove Chasm Trap



Property PA14 is at the branch B007

Contents

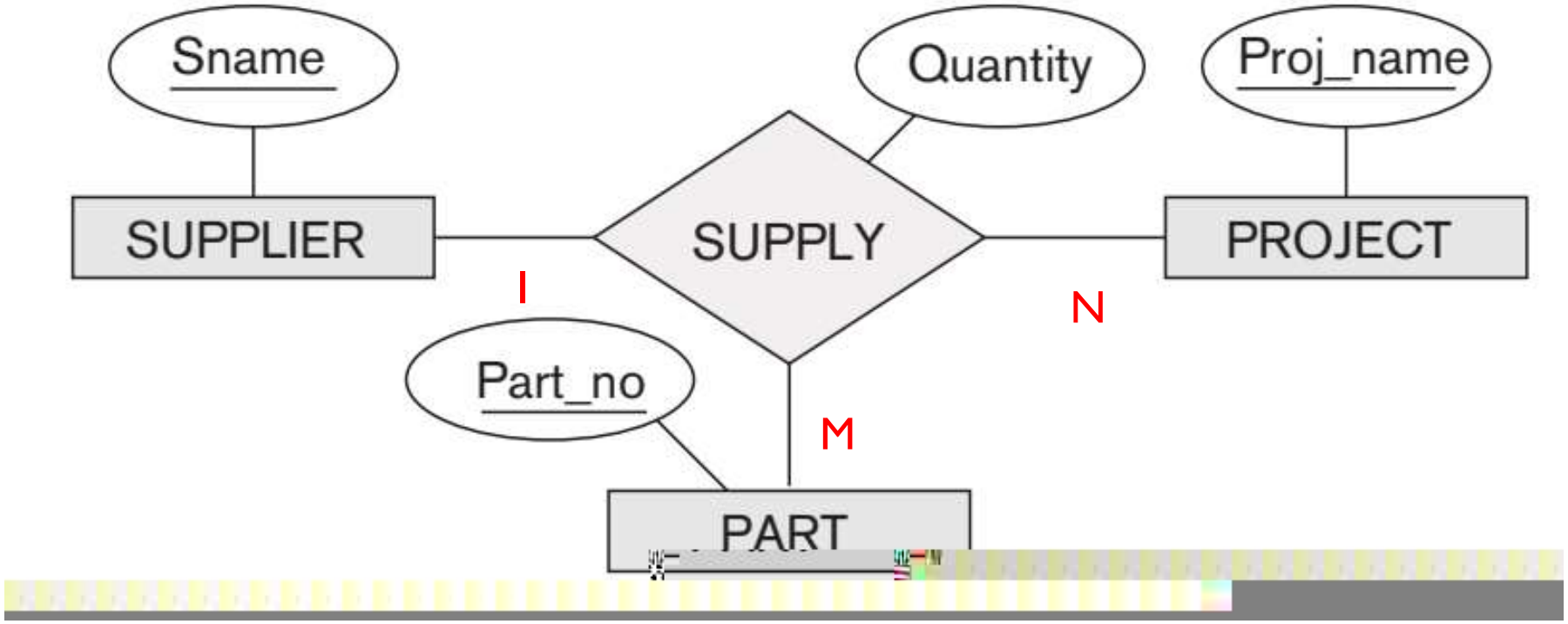

Q
A

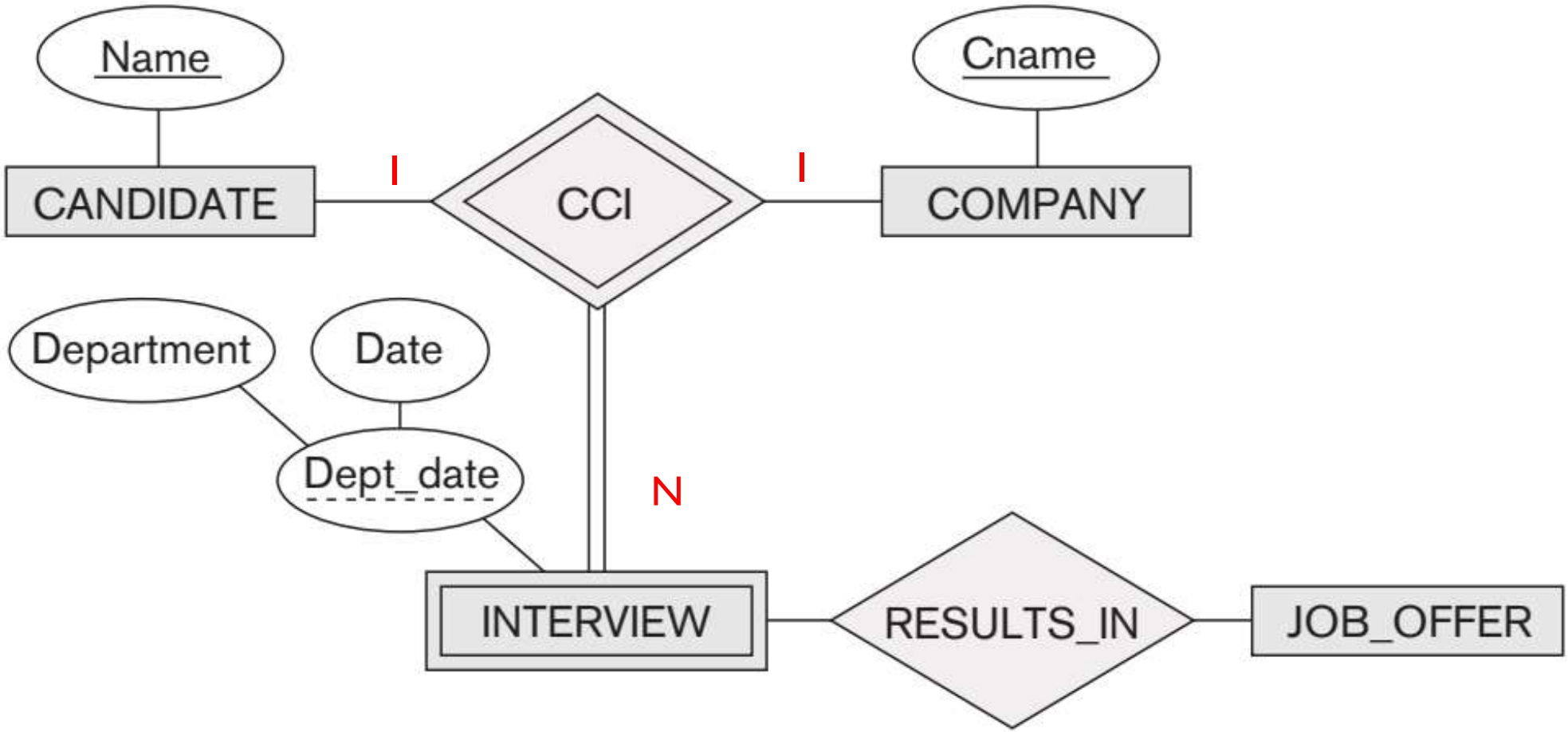
Exercise 1: University Database

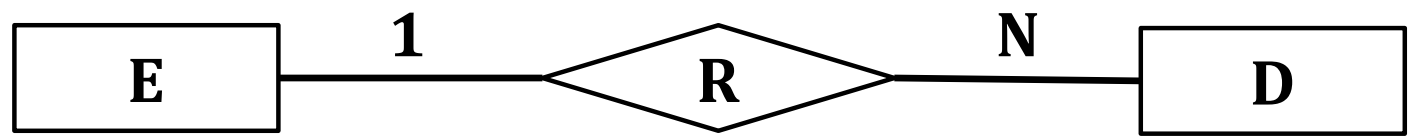
The university database maintains records of its departments, lecturers, course modules, and students. The university consists of departments. Each department has a unique name and some other descriptive attributes. A department must also have a number of lecturers, one of which is the head of department. All lecturers have different names (we assume so anyway). They must teach one or more modules. A lecturer can only belong to one department. Modules are offered by departments . A module is taught by one lecturer. They must also be attended by some students. Each module has a unique module number. Students must enrol for a number of modules. Each student is given a unique student number

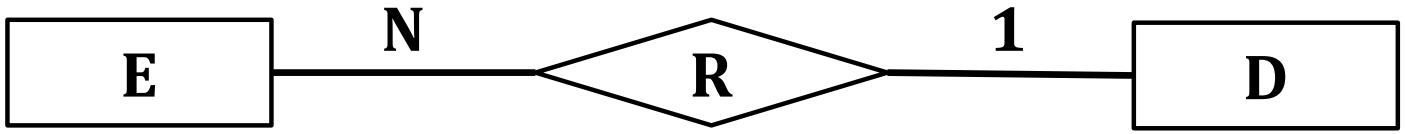
Exercise 2: Small LIB Database

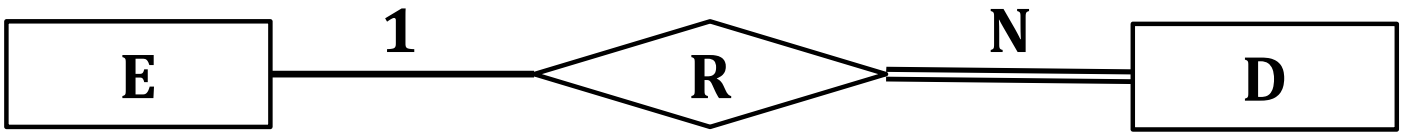
You are to design a database for a small library. The database needs to store data about various branches and about books the library holds. Each branch has an id (unique), name (unique) and an address. For each book, the database should record the book id (unique), title, publisher and the year of publication. A book may have several authors, and each author is represented by his/her name. A book typically has several copies. Each copy of a book is given a copy number. The availability of a book should be known, as well as the total number of copies

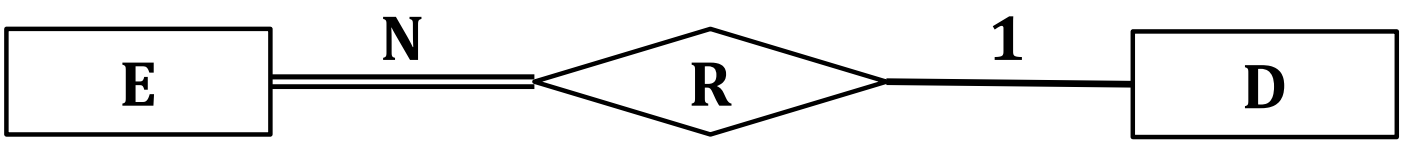

Sname
Quantity
Proj_name
SUPPLIER
SUPPLY
PROJECT
1
N
Part_no
M
PART

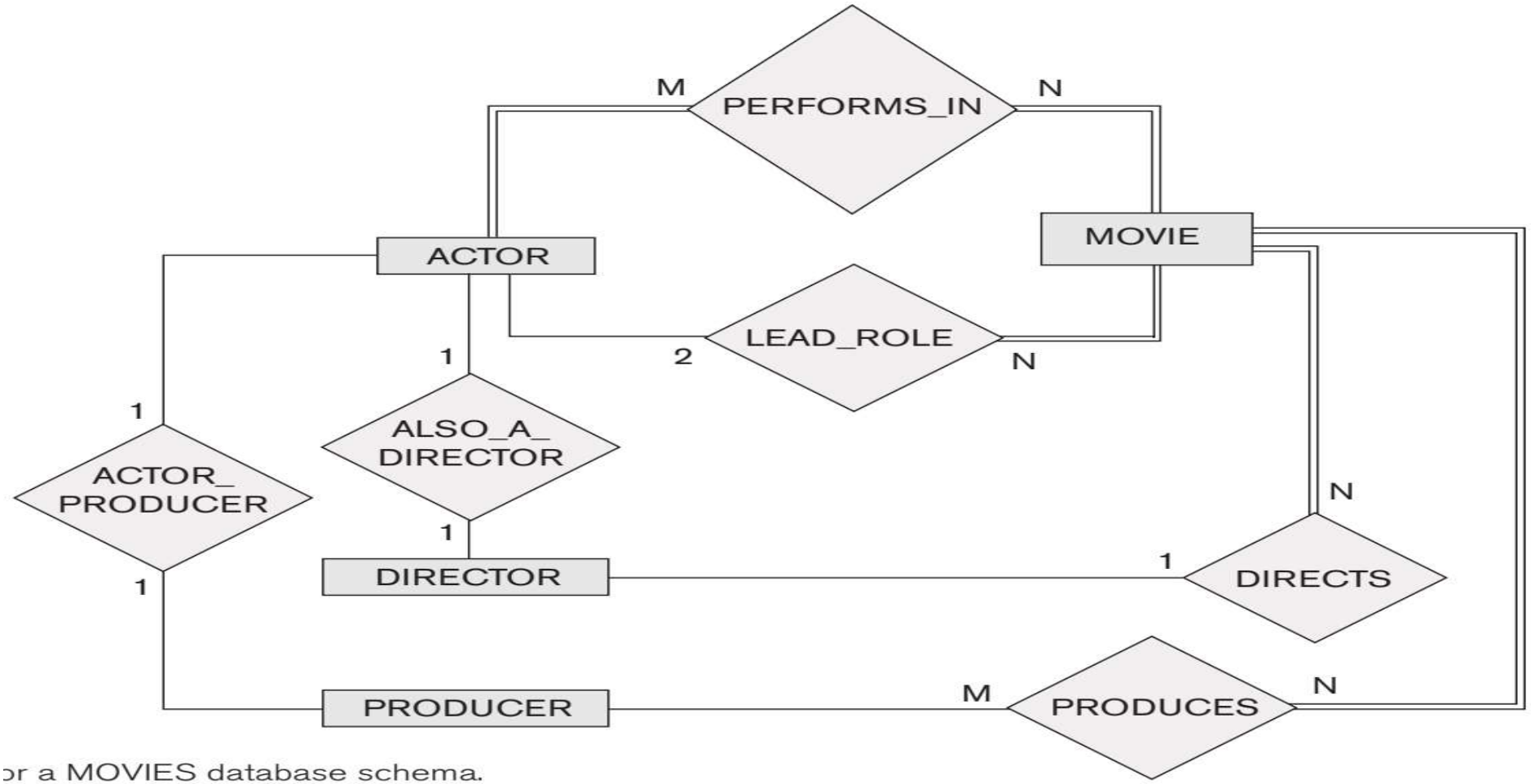

Name
Cname
CANDIDATE
1
CCI
1
COMPANY
Department
Date
Dept_date
N
INTERVIEW
RESULTS_IN
JOB_OFFER


E
1
R
N
D


E
N
R
1
D

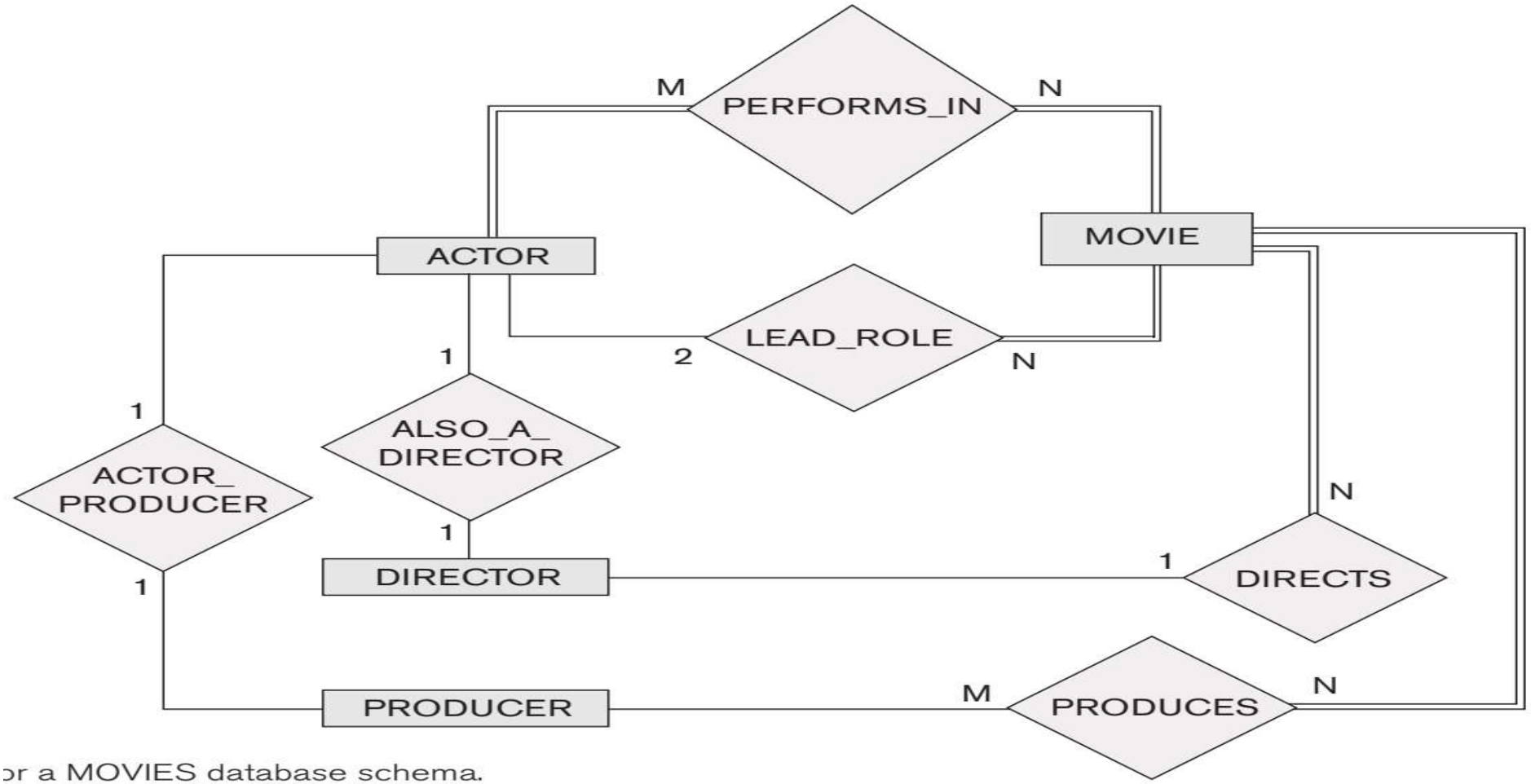

E
1
R
N
D


E
N
R
1
D



or a MOVIES database schema.

a. There are no actors in this database that have been in no movies.
b. There are some actors who have acted in more than ten movies.
c. Some actors have done a lead role in multiple movies.
d. A movie can have only a maximum of two lead actors.



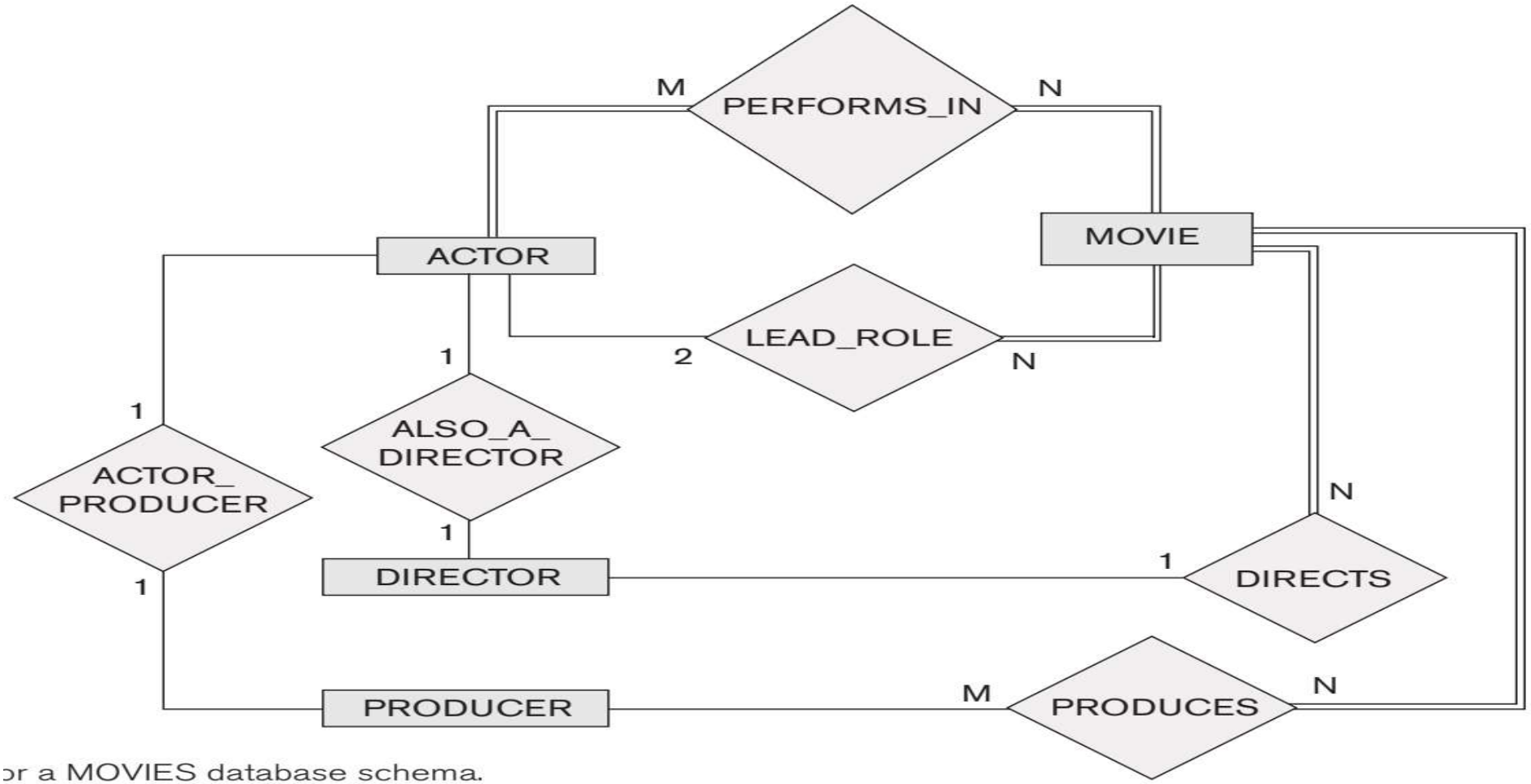
or a MOVIES database schema.

e. Every director has been an actor in some movie.

f. No producer has ever been an actor.

g. A producer cannot be an actor in some other movie.

h. There are movies with more than a dozen actors.



or a MOVIES database schema.

i. Some producers have been a director as well.

j. Most movies have one director and one producer.

k. Some movies have one director but several producers.

l. There are some actors who have done a lead role, directed a movie, and produced a movie.

m. No movie has a director who also acted in that movie.

Case study: Requirements - Collection & Analysis

Data requirements

- Entities
- Attributes
- Relationships
- Constraints

GROUP A	GROUP B
A system for course registration of HCMUT	An order management system for a shop
	

Case study: Requirements - Collection & Analysis

Data requirements

- Entities
- Attributes
- Relationships
- Constraints

GROUP A	GROUP B
A system for course registration of HCMUT	A system for a Library of a University

Case study: Identify Entity Types and Attributes

GROUP A	GROUP B
A system for course registration of HCMUT	An order management system for a shop

Entity type

Simple att.

Key

Multi-valued attribute

Composite att.

Derived attribute

Case study: Identify Entity Types and Attributes

GROUP A	GROUP B
A system for course registration of HCMUT	A system for a Library of a University

Entity type

Simple att.

Key

Multi-valued attribute

Composite att.

Derived attribute