Chapter Four

Database Design

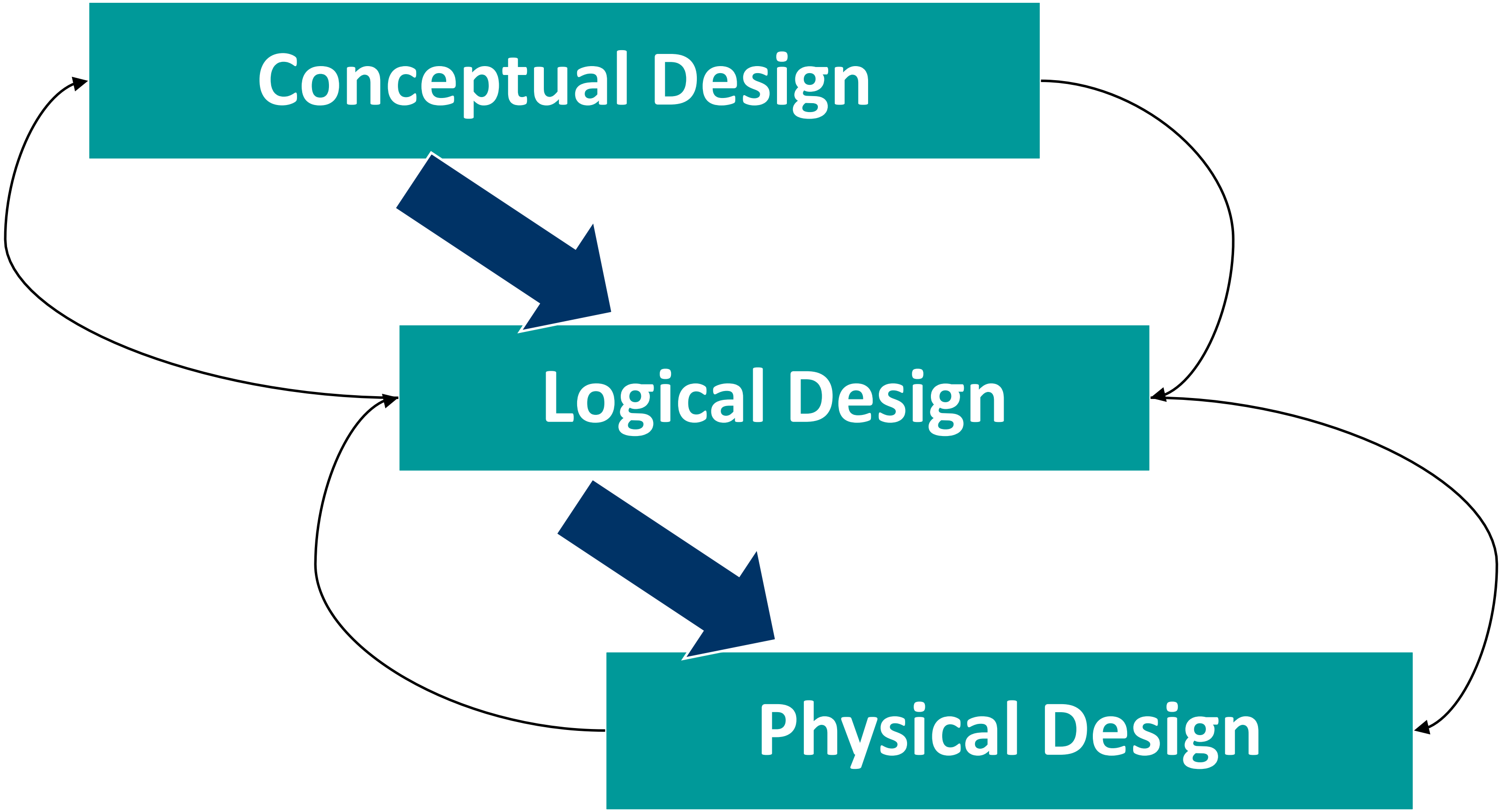
Sub phases

1. Conceptual Database Design
2. Logical Design Database, and
3. Physical Database Design

Introduction

- In developing a good design, one should answer such questions as:
  - What are the relevant **Entities** for the Organization
  - What are the important **features of each Entity**
  - What are the important **Relationships**
  - What are the important **queries** from the user
  - What are the other **requirements** of the Organization and the Users

Levels of Database Design



Conceptual Database Design

- Identify what are the **entities/entity types**
- Identify what are the **attributes**
- Identify **relationship types**
- Identify what are the **constraints/business rules** that hold
- Draw **Entity-Relationship Diagram (ERD)**
- Review the conceptual data model with user

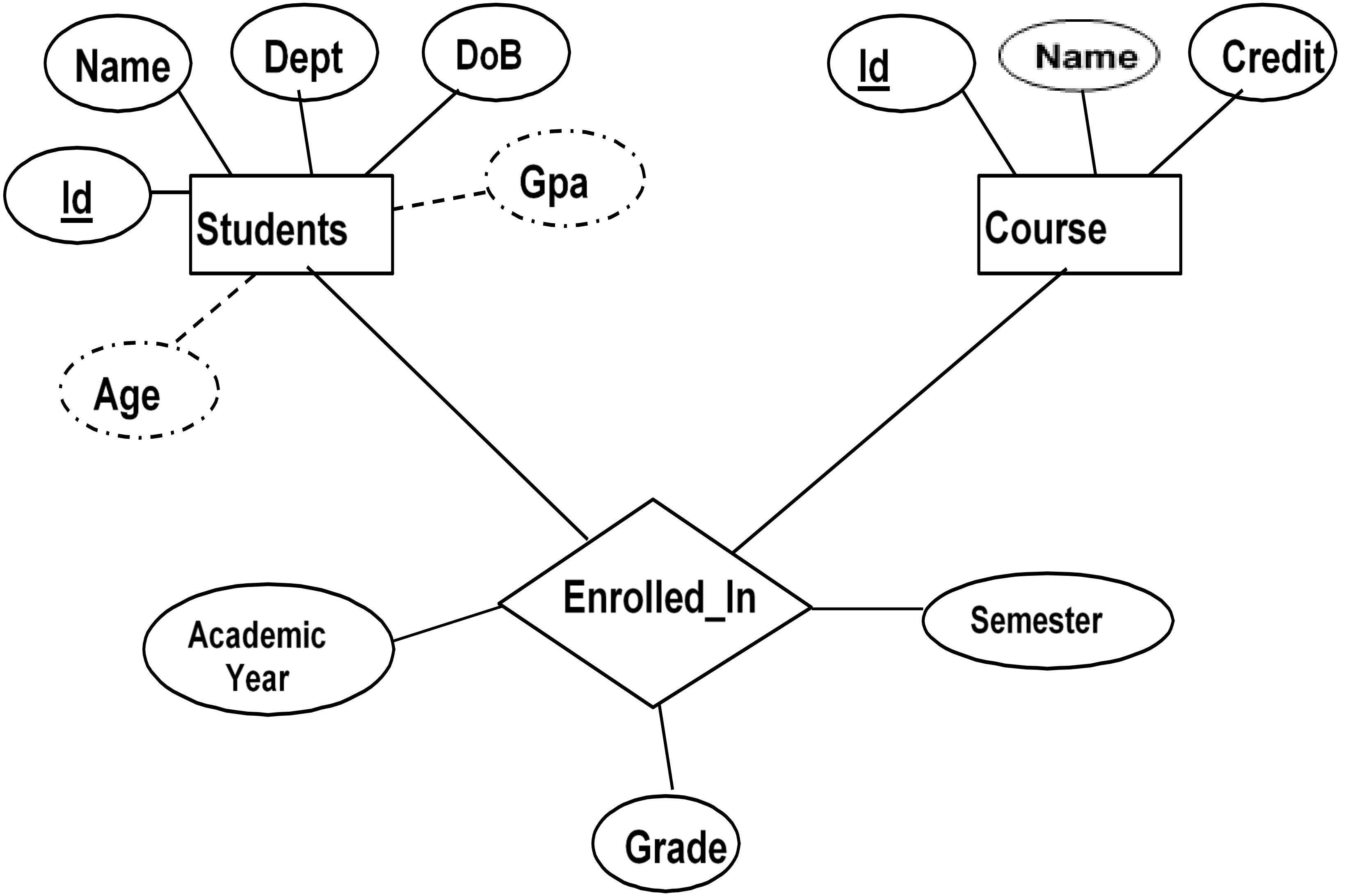
Developing an E-R Diagram(1)

- Information gathered by
  - Interviewing end users individually and in a group
  - Questionnaire survey
  - Direct observation
  - Examining different documents

Developing an E-R Diagram(2)

A student record management system will have the following two basic data object categories with their own features or properties: Students will have an Id, Name, Dept, Age, GPA and Course will have an Id, Name, Credit Hours. Whenever a student enroll in a course in a specific Academic Year and Semester, the Student will have a grade for the course.

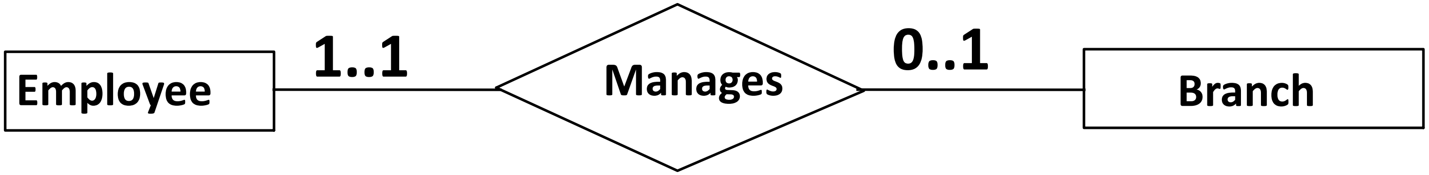
Developing an E-R Diagram(3)



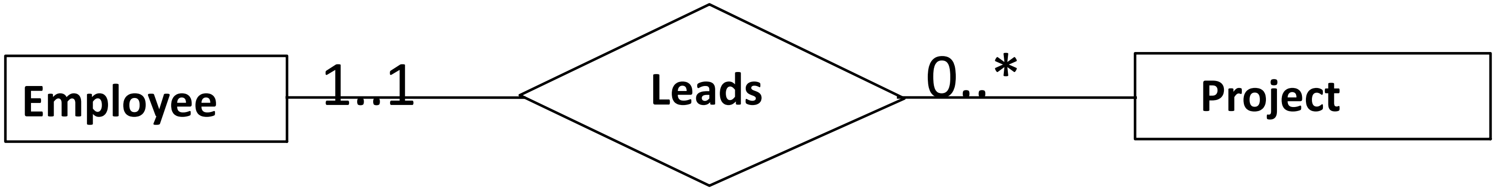
Relationship cardinality

The multiplicity of the relationship:

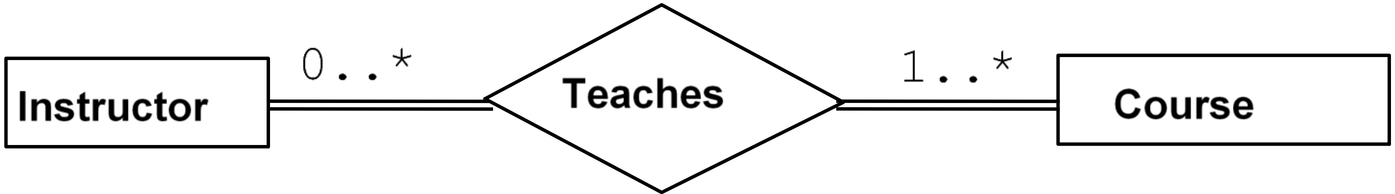
- One branch can only have one manager.
- One employee could manage either one or no branches



- One staff may Lead one or more project(s)
- One project is Lead by one staff



- One Instructor Teaches one or more Course(s)
- One Course Thought by Zero or more Instructor(s)



Participation constraints(1)

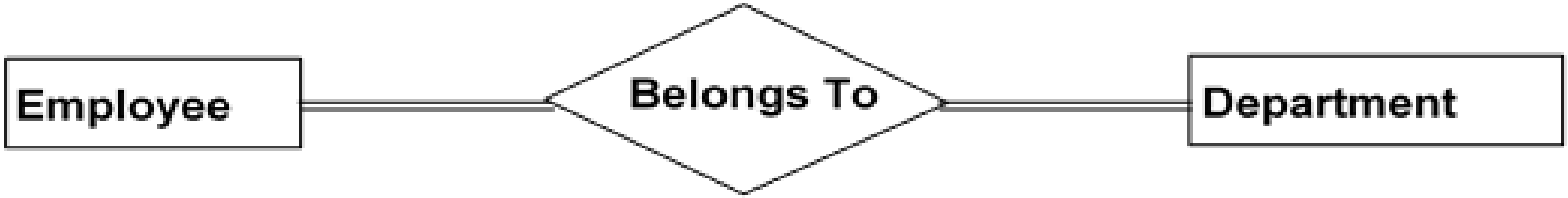
- **Total participation**
  - every tuple in the entity or relation participates in at least one relationship by taking a role
  - This means, every tuple in a relation will be attached with at least one other tuple.
  - The entity with total participation in a relationship will be connected to the relationship using a **double line**.

Participation constraints(2)

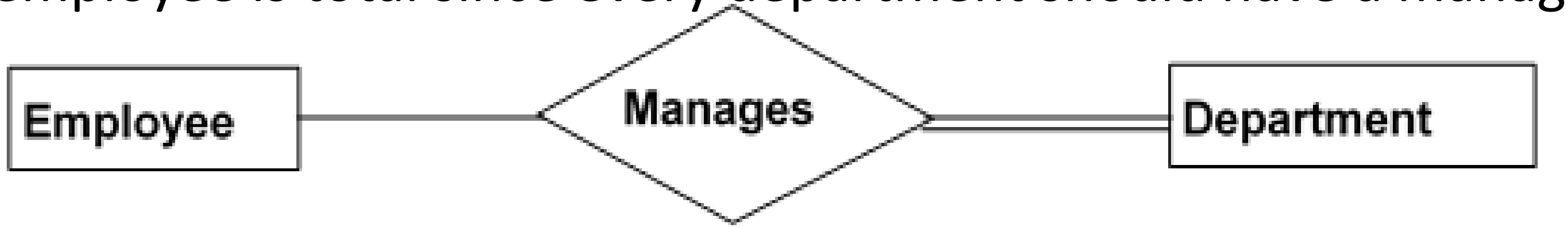
- **Partial participation**
  - some tuple in the entity or relation may not participate in the relationship.
  - This means, there is at least one tuple from that Relation not taking any role in that specific relationship.
  - The entity with partial participation in a relationship will be connected to the relationship using a **single line.**

Participation constraints(3)

- Participation of EMPLOYEE in ‘**belongs to’** relationship with DEPARTMENT is total since every employee should belong to a department.
- Participation of DEPARTMENT in **‘belongs to’** relationship with EMPLOYEE is total since every department should have more than one employee



- Participation of employee in **‘manages’** relationship with Department, is partial participation since not all employees are managers
- Participation of department in **‘Manages’** relationship with employee is total since every department should have a manager.

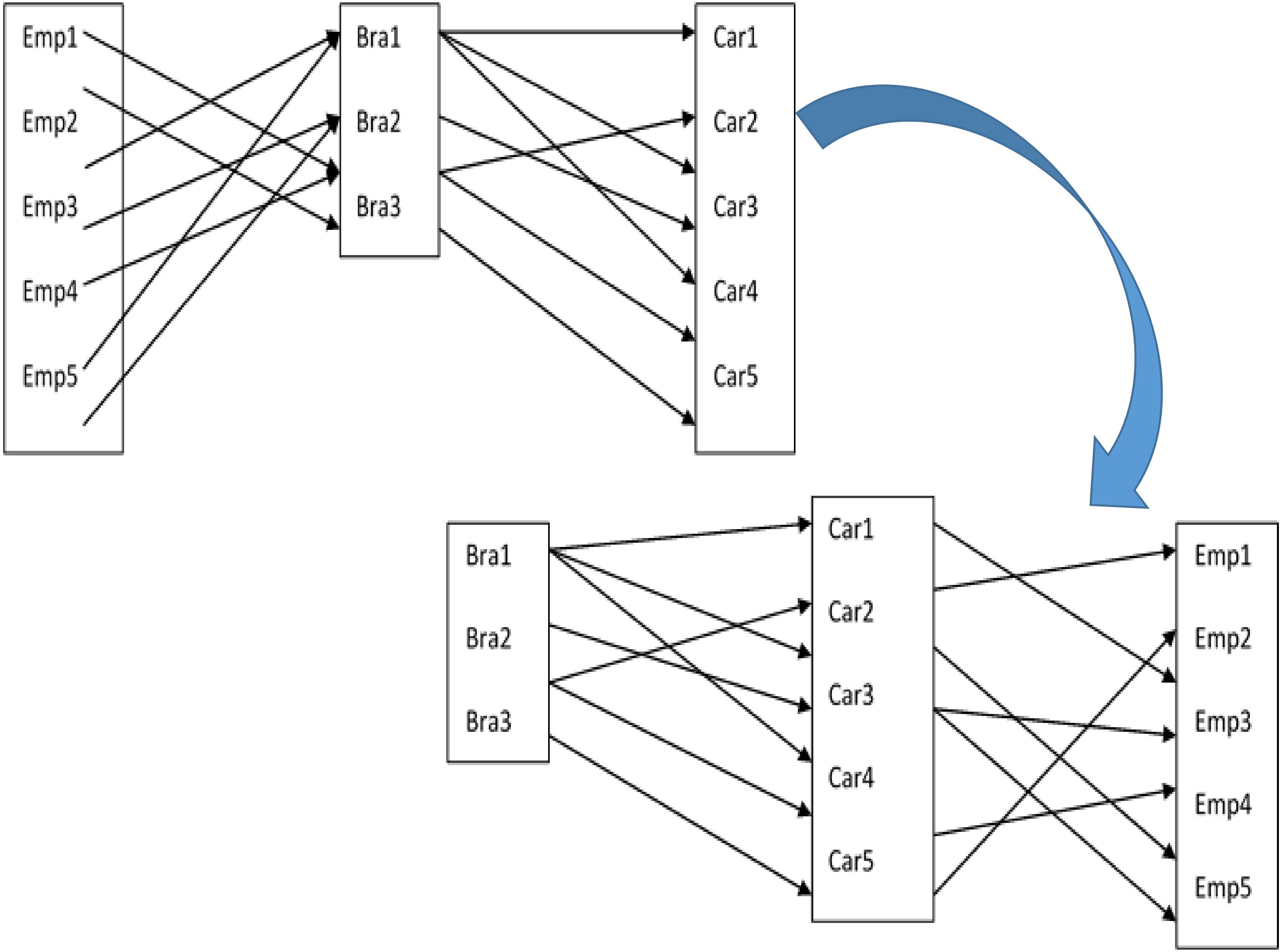


Problem in ER Modeling

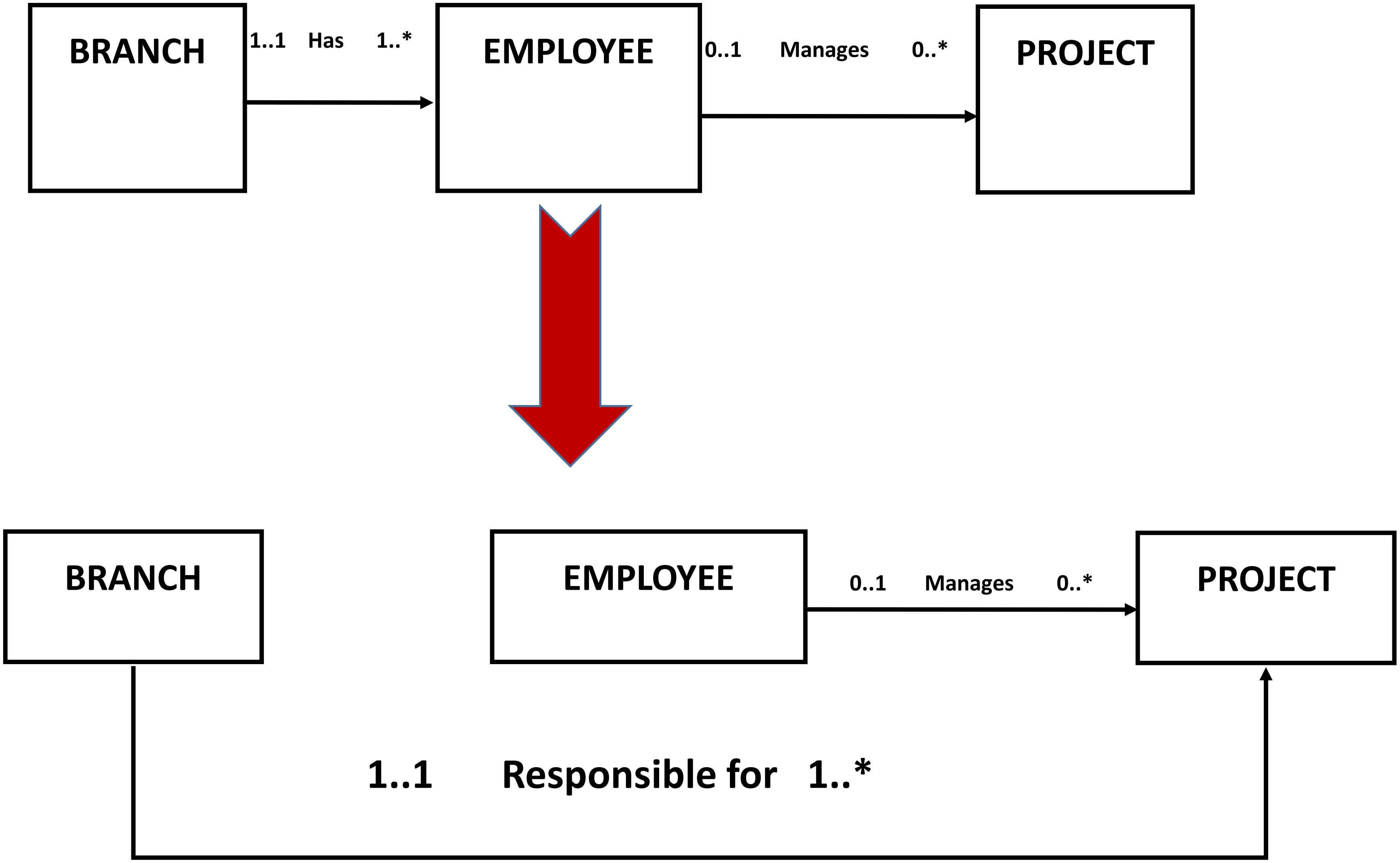
- **Connection traps** are problems arising from misinterpreting certain relationships.

1. **Fan trap**
   - Occurs where a model represents a relationship between entity types, but the pathway between certain entity occurrences is **ambiguous.**
   - **Solution: restructuring**
2. **Chasm Trap**
   - Occurs where a model suggests the existence of a relationship between entity types, but the path way does not exist between certain entity occurrences.
   - **Solution:** to add another relationship between the extreme entities.

Fan trap


Emp1
Emp2
Emp3
Emp4
Emp5
Bra1
Bra2
Bra3
Car1
Car2
Car3
Car4
Car5
Bra1
Bra2
Bra3
Car1
Car2
Car3
Car4
Car5
Emp1
Emp2
Emp3
Emp4
Emp5

Chasm Trap


BRANCH
1..1
Has
1..*
EMPLOYEE
0..1
Manages
0..*
PROJECT
BRANCH
EMPLOYEE
0..1
Manages
0..*
PROJECT
1..1
Responsible for
1..*

Enhanced ER (EER) Model

- **EER Concepts**
  - Generalization
  - Specialization
  - Sub classes
  - Super classes
  - Attribute Inheritance
  - Constraints on Specialization and Generalization

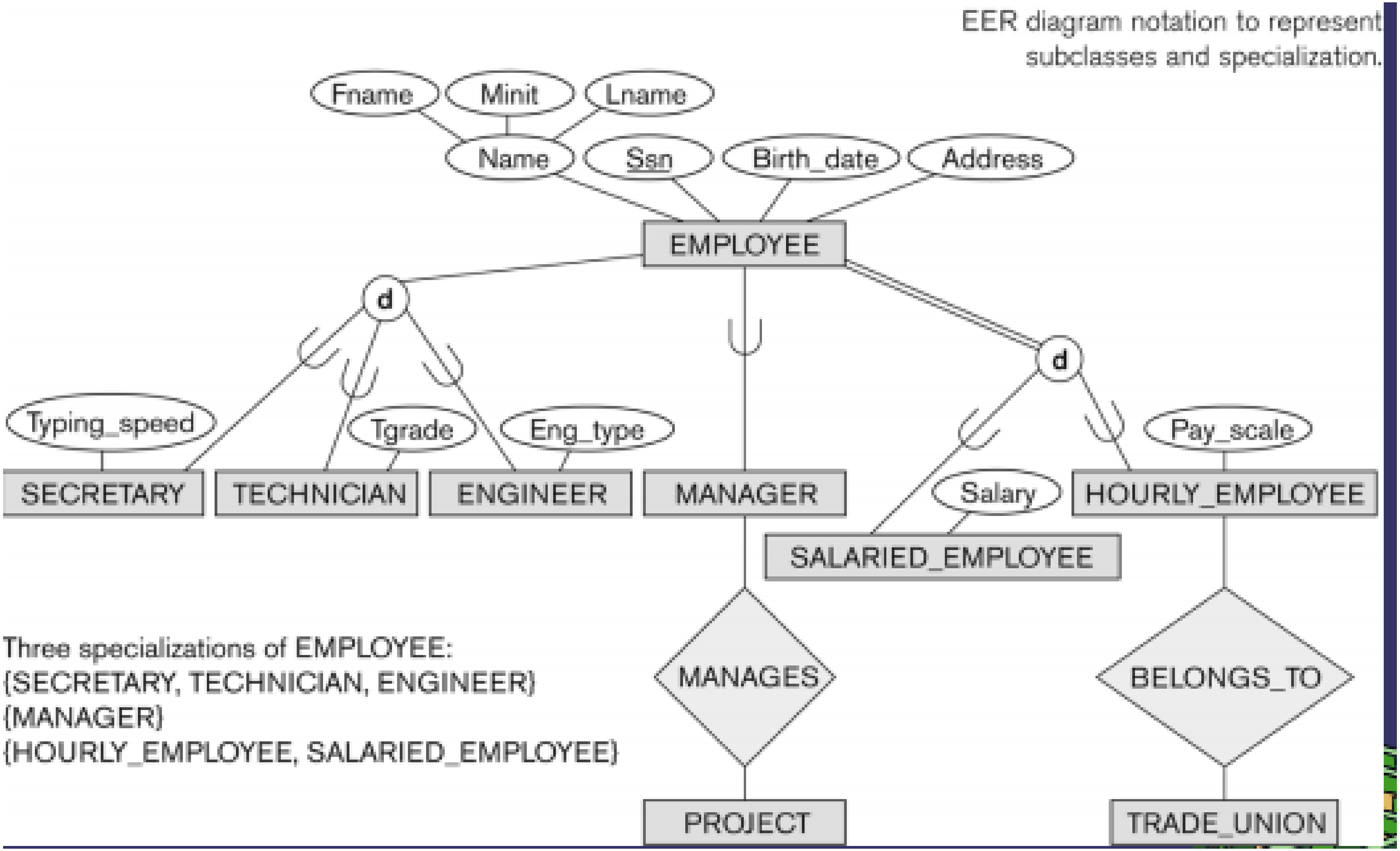
Subclass and superclass(1)

- An entity type may have additional meaningful subgroupings of its entities
- Example: EMPLOYEE may be further grouped into:
  - Based on the EMPLOYEE’s Job (MANAGER, SECRETARY, ENGINEER, TECHNICIAN,)
  - Based on the EMPLOYEE’s method of pay (SALARIED_EMPLOYEE, HOURLY_EMPLOYEE)
- EER diagrams extend ER diagrams to represent these additional subgroupings, called subclasses or subtypes

Subclass and superclass(2)

- Each of these subgroupings is a subset of EMPLOYEE entities
- Each is called a subclass of EMPLOYEE
- EMPLOYEE is the superclass for each of these subclasses
- These are called superclass/subclass relationships:
  - EMPLOYEE/SECRETARY
  - EMPLOYEE/TECHNICIAN
  - EMPLOYEE/MANAGER
- Subclass entity inherits all attributes and relationships of superclass

Subclass and superclass(2)



EER diagram notation to represent subclasses and specialization.

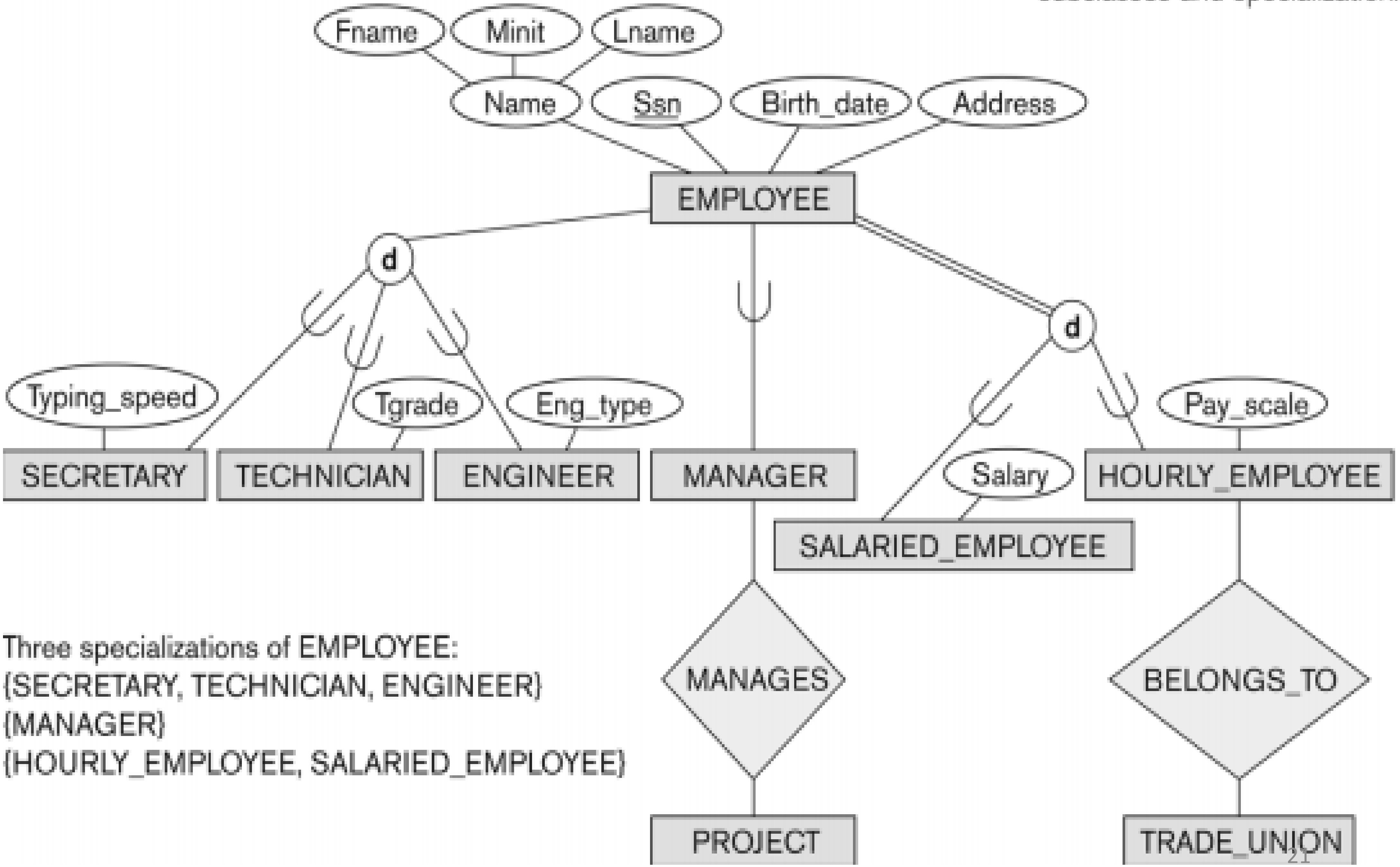
Three specializations of EMPLOYEE:
{SECRETARY, TECHNICIAN, ENGINEER}
{MANAGER}
{HOURLY_EMPLOYEE, SALARIED_EMPLOYEE}

Specialization(1)

- Specialization is the process of defining a set of subclasses of a superclass
- The set of subclasses is based upon some distinguishing characteristics of the entities in the superclass
- Example: {SECRETARY, ENGINEER, TECHNICIAN} is a specialization of EMPLOYEE based upon **job type**.
- May have several specializations of the same superclass
- Example: Another specialization of EMPLOYEE based on method of **pay** is {SALARIED_EMPLOYEE, HOURLY_EMPLOYEE}
- Attributes of a subclass are called specific or **local attributes**. For example, the attribute TypingSpeed of SECRETARY
- The subclass can also participate in specific relationship types. For example, a relationship BELONGS_TO of HOURLY_EMPLOYEE

Specialization(2)

EER diagram notation to represent subclasses and specialization.

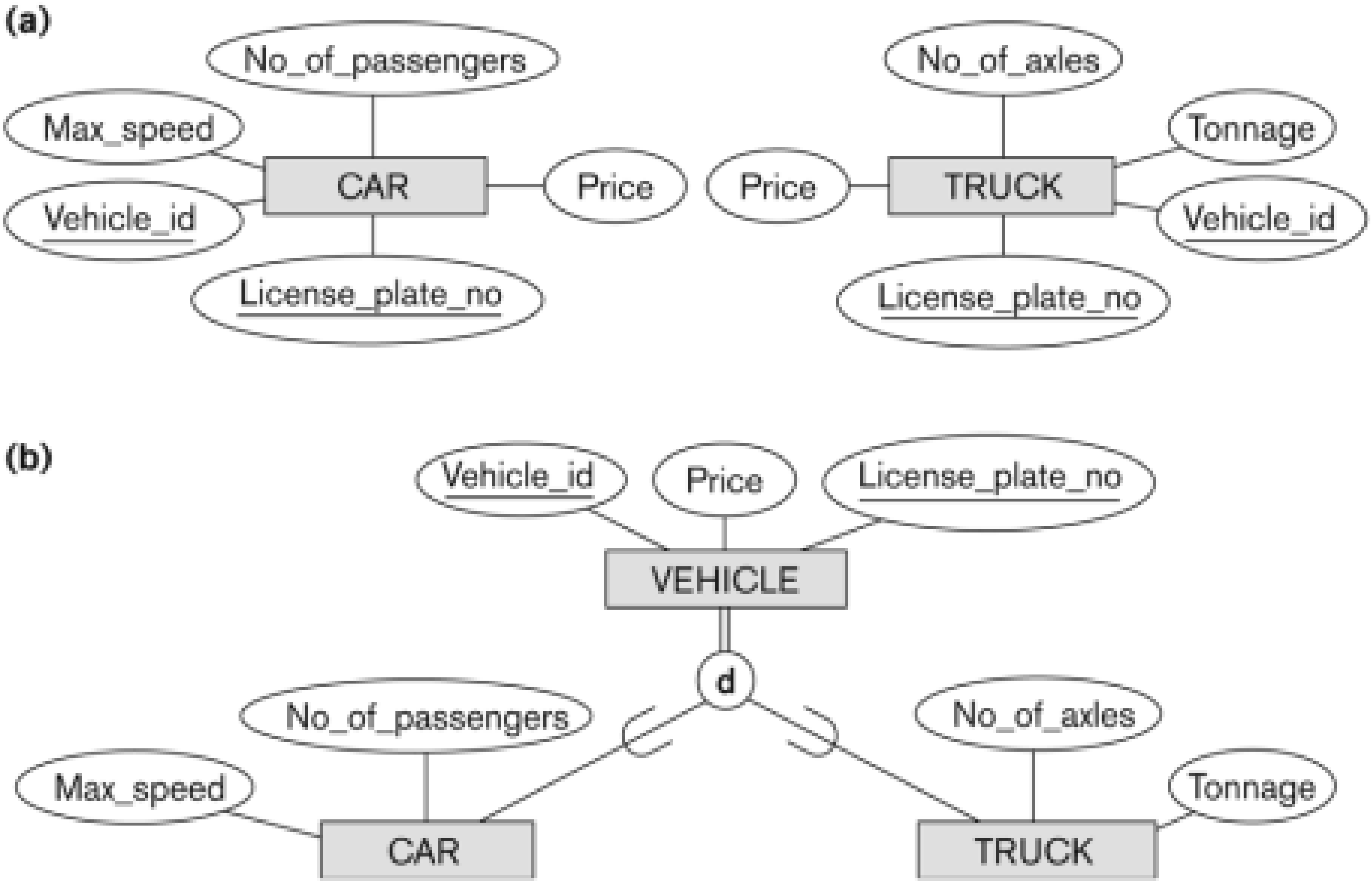


Three specializations of EMPLOYEE:
{SECRETARY, TECHNICIAN, ENGINEER}
{MANAGER}
{HOURLY_EMPLOYEE, SALARIED_EMPLOYEE}

Generalization(1)

- Generalization is the reverse of the specialization process
- Several classes with common features are generalized into a superclass
- Original classes become its subclasses
  - Example: CAR, TRUCK generalized into VEHICLE
  - Both CAR, TRUCK become subclasses of the superclass VEHICLE
- We can view {CAR, TRUCK} as a specialization of VEHICLE
- Alternatively, we can view VEHICLE as a generalization of CAR and TRUCK

Generalization (2)



Generalization. (a) Two entity types, CAR and TRUCK.
(b) Generalizing CAR and TRUCK into the superclass VEHICLE.

Constraints on Specialization and Generalization (1)

- **Disjointness Constraint (Disjoint or Overlapped)**
  - Specifies: subclasses of the specialization must be disjoint
  - An entity can be a member of at most one of the subclasses of the specialization Specified by **d** in EER diagram
  - If not disjoint, specialization is overlapping
  - That is the same entity may be a member of more than one subclass of the specialization: Specified by **o** in EER diagram
  - **Disjoint**: an entity can belong to only one lower level entity set
  - **Overlapping:** an entity can belong to more than one lower level entity set

Constraints ……. (2)

- **Completeness Constraint (Total and Partial)**
  - May be total or partial
  - Total specifies that every entity in the superclass must be a member of some subclass in the specialization/generalization
    - Shown in EER diagrams by a **double line**
  - Partial allows an entity not to belong to any of the subclasses
    - Shown in EER diagrams by a single line
- **Hence we have four possible constraints**
  - Disjoint AND Total
  - Disjoint AND Partial
  - Overlapping AND Total
  - Overlapping AND Partial
- Note: Generalization is usually total because the superclass is derived from the subclasses

Thanks !!!