An introduction to ER model weaknesses & EER Model

Assoc. Prof. Dr. DANG TRAN KHANH

Outline

- ERD exercises
- Introduction to ER model weaknesses
- Enhanced/Extended ER model
- Reading:
  - [1]: Chapter 4 (7th ed)

ERD exercises

- Exercise 1: review …
- Bonus marks if have a good solution for each of the following problems
  - ERD design

Exercises

- (Given in the class)

Outline

- ERD exercises
- Introduction to ER model weaknesses
- Enhanced/Extended ER model
- Reading:
  - [1]: Chapter 4 (7th ed)

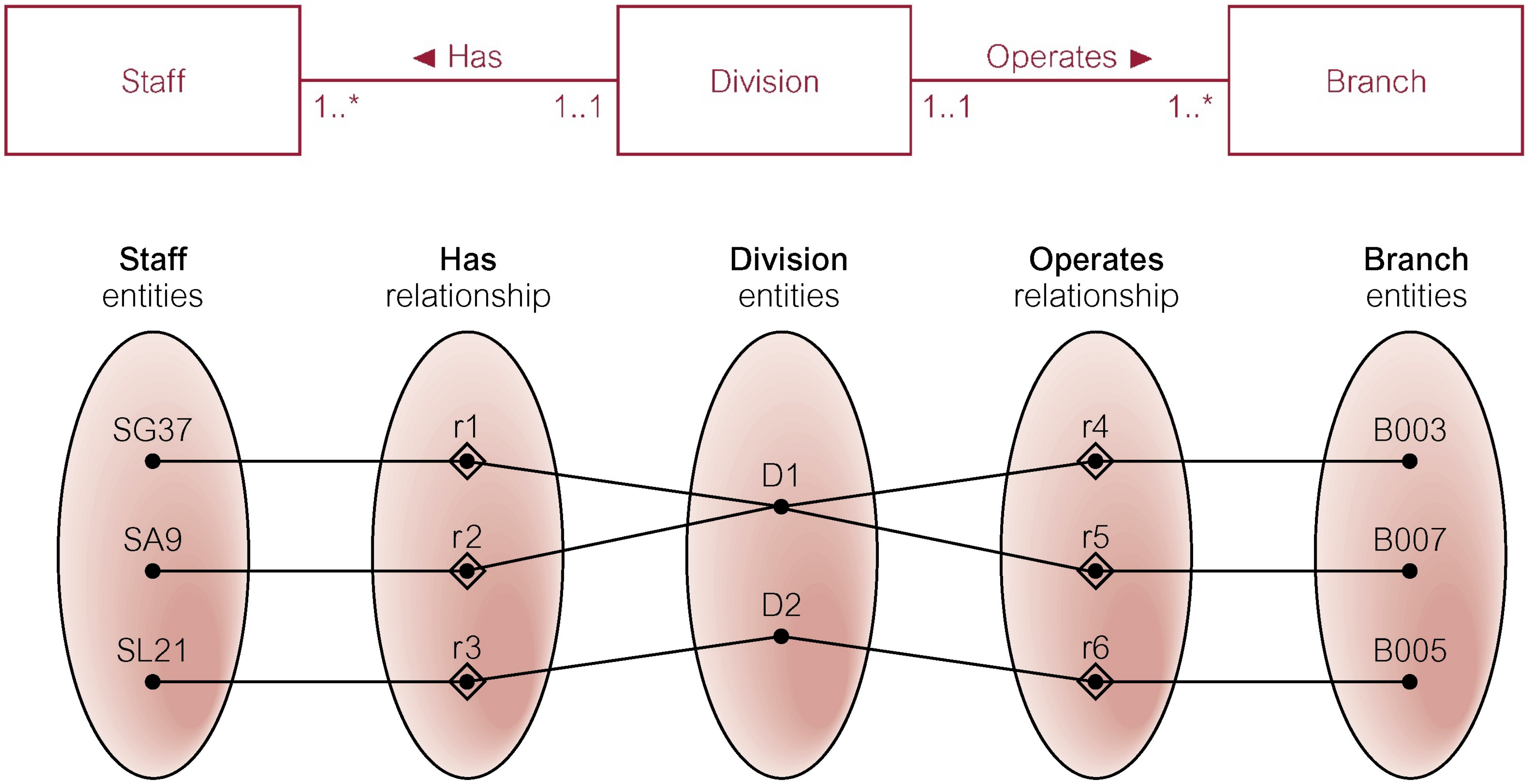
Problems with ER Models

- 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*

Problems with ER Models

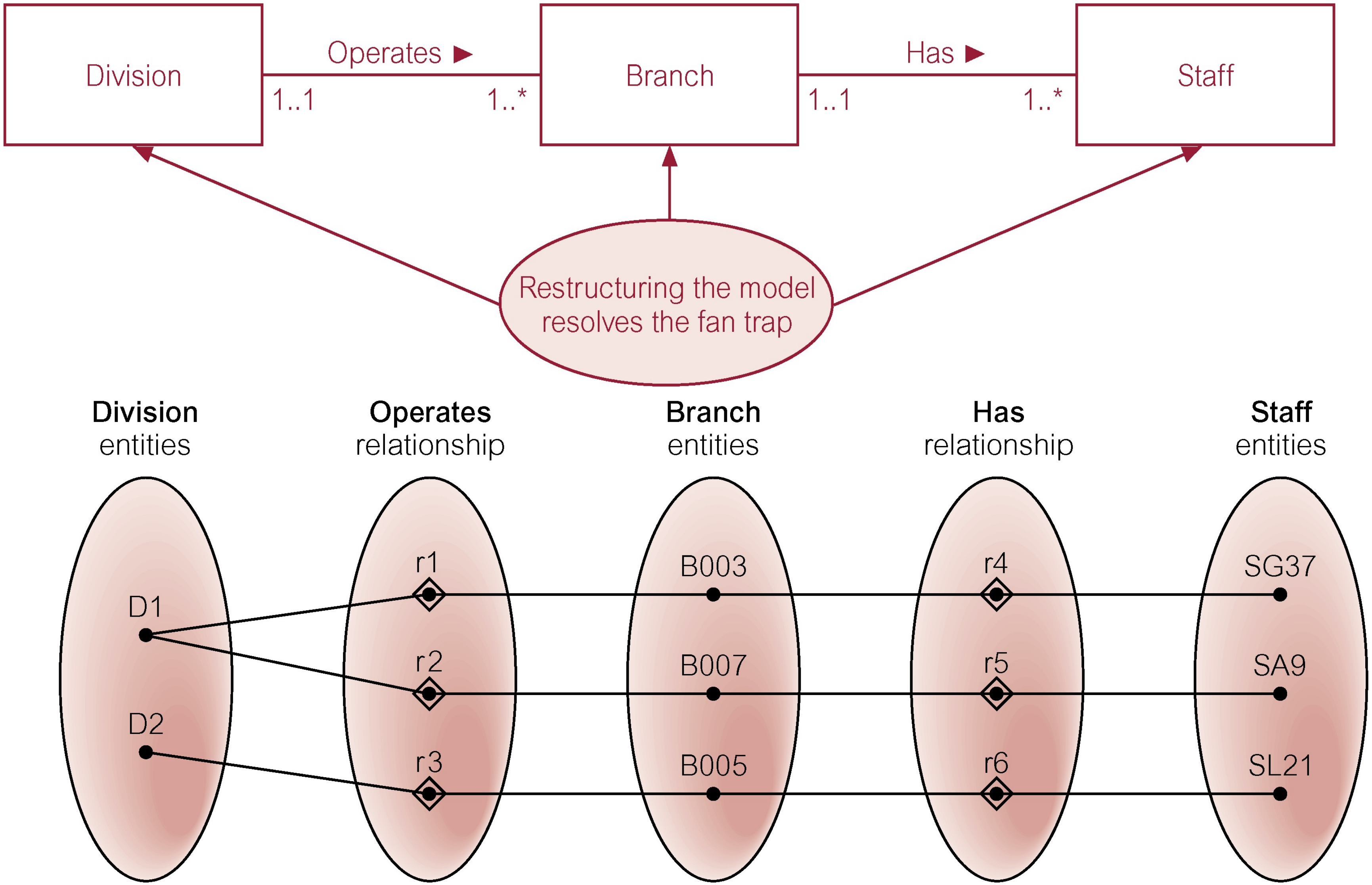
- Fan Trap
  - Where a model represents a relationship between entity types, but pathway between certain entity occurrences is ambiguous
  - Usually: two or more 1:N relationships fan out from the same entity
- Chasm Trap
  - Where a model suggests the existence of a relationship between entity types,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

An Example of a Fan Trap



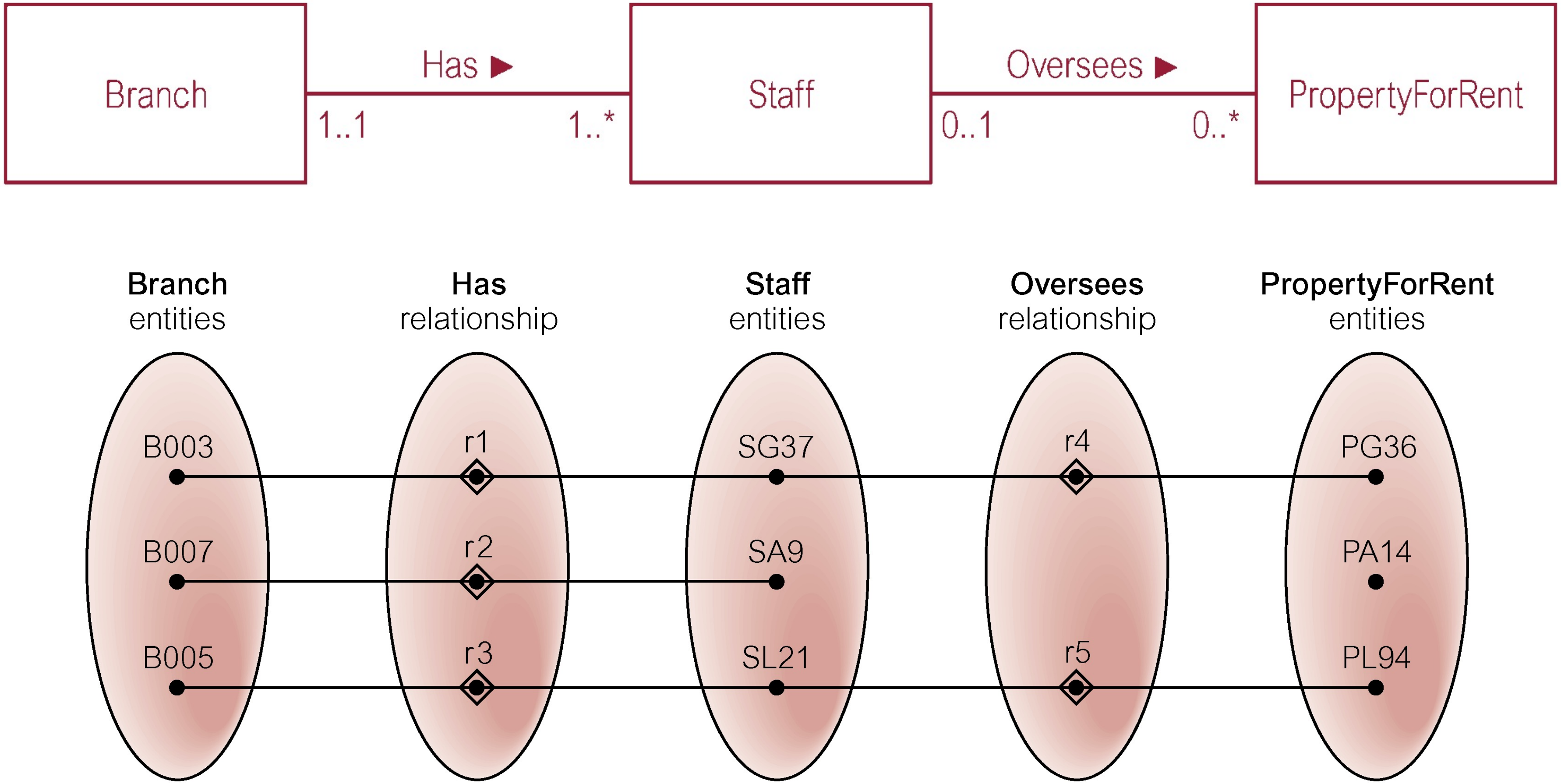
At which branch office does staff number SG37 work?

Restructuring ER model to remove Fan Trap



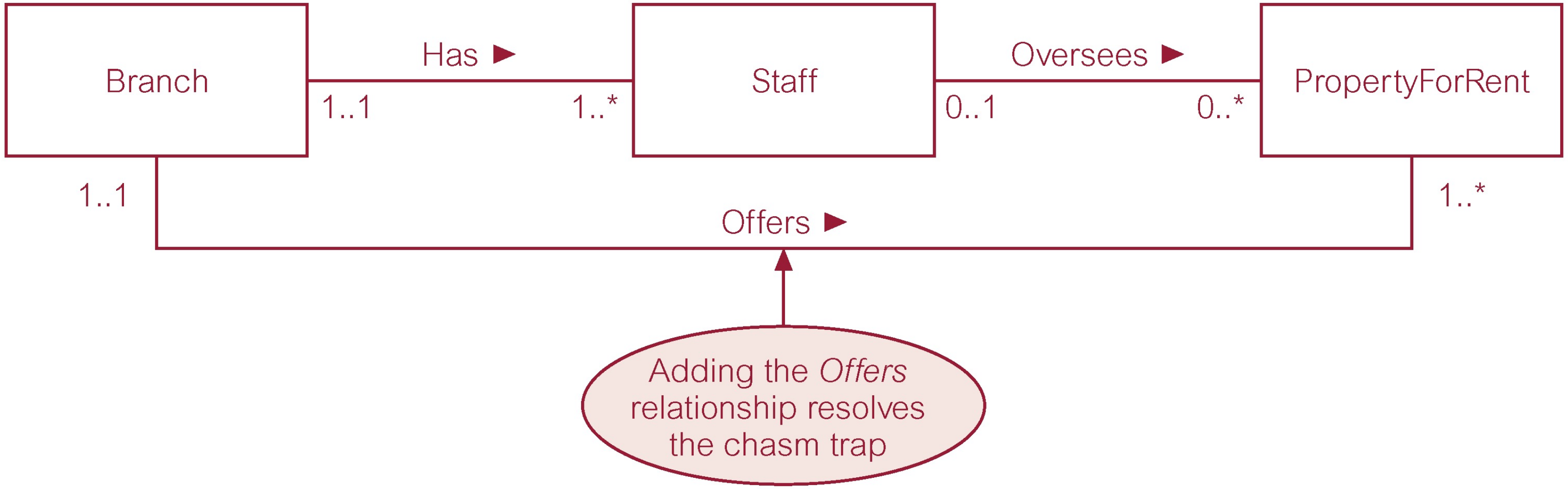
SG37 works at branch B003

An Example of a Chasm Trap

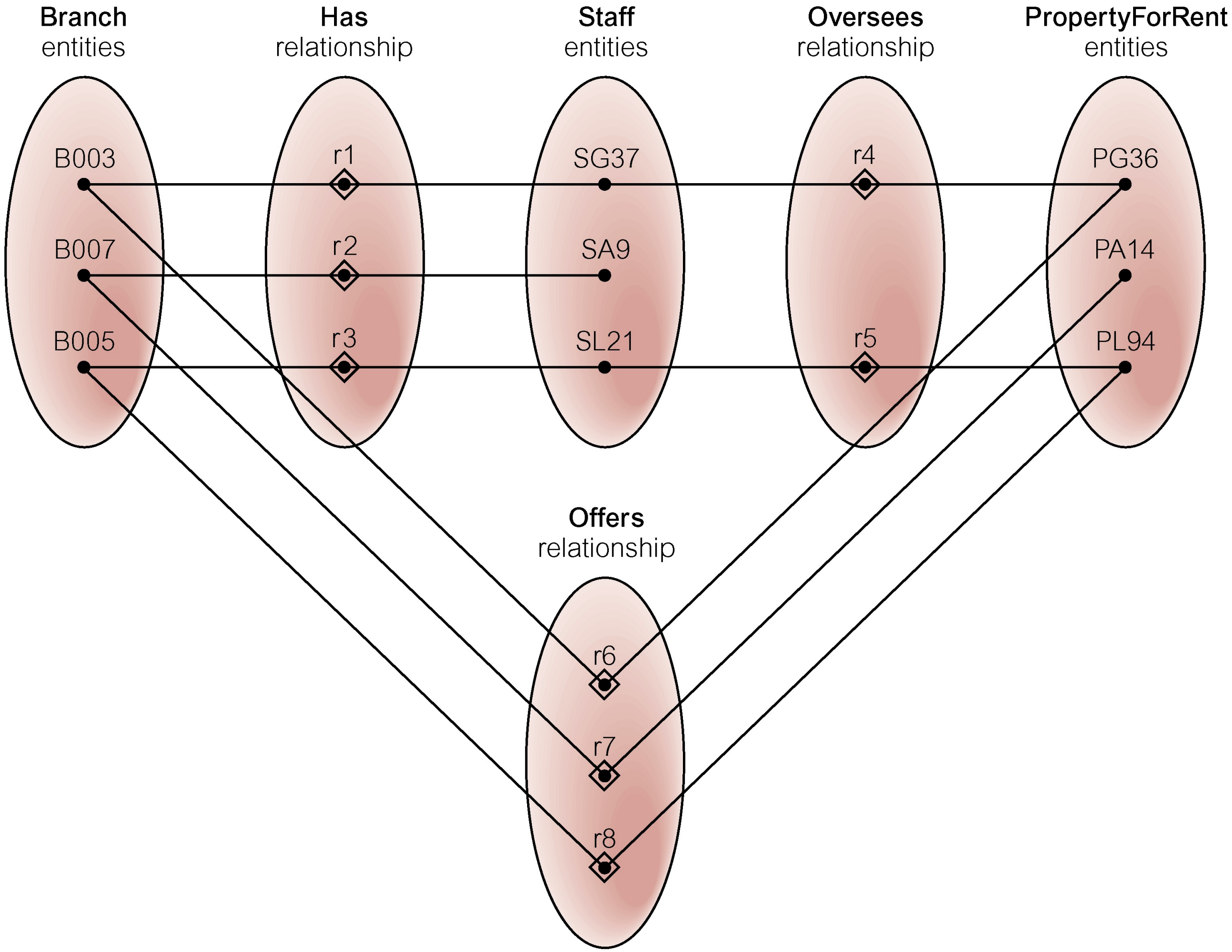


At which branch office is property PA14 available?

ER Model restructured to remove Chasm Trap



ER Model restructured to remove Chasm Trap



Problems with ER Models

- Many semantic constraints can not be expressed by ER model
  - E.g.: The changing of the salary must be an increase ☺
- Further reading: A. Badia: ”Entity-Relationship Modeling Revisited”, SIGMOD Record, 33(1), March 2004, 77-82

Outline

- ERD exercises
- Introduction to ER model weaknesses
- Enhanced/Extended ER model
- Reading:
  - [1]: Chapter 4 (7th ed)

EER model

- Limitations of Basic Concepts of the ER Model
- Enhanced-ER (EER) Model Concepts
- Subclasses and Superclasses
- Specialization and Generalization
- Specialization / Generalization Hierarchies, Lattices and Shared Subclasses
- Categories
- Formal Definitions of EER Model
- Database Design Modeling Tools (seminar in w4)

Limitations of Basic Concepts of the ER model

- Since the 1980s there has been an increase in emergence of new database applications with more demanding requirements
- Basic concepts of ER modelling are not sufficient to represent requirements of newer, more complex applications (cf. Web for more details)
- Response is development of additional 'semantic' modeling concepts

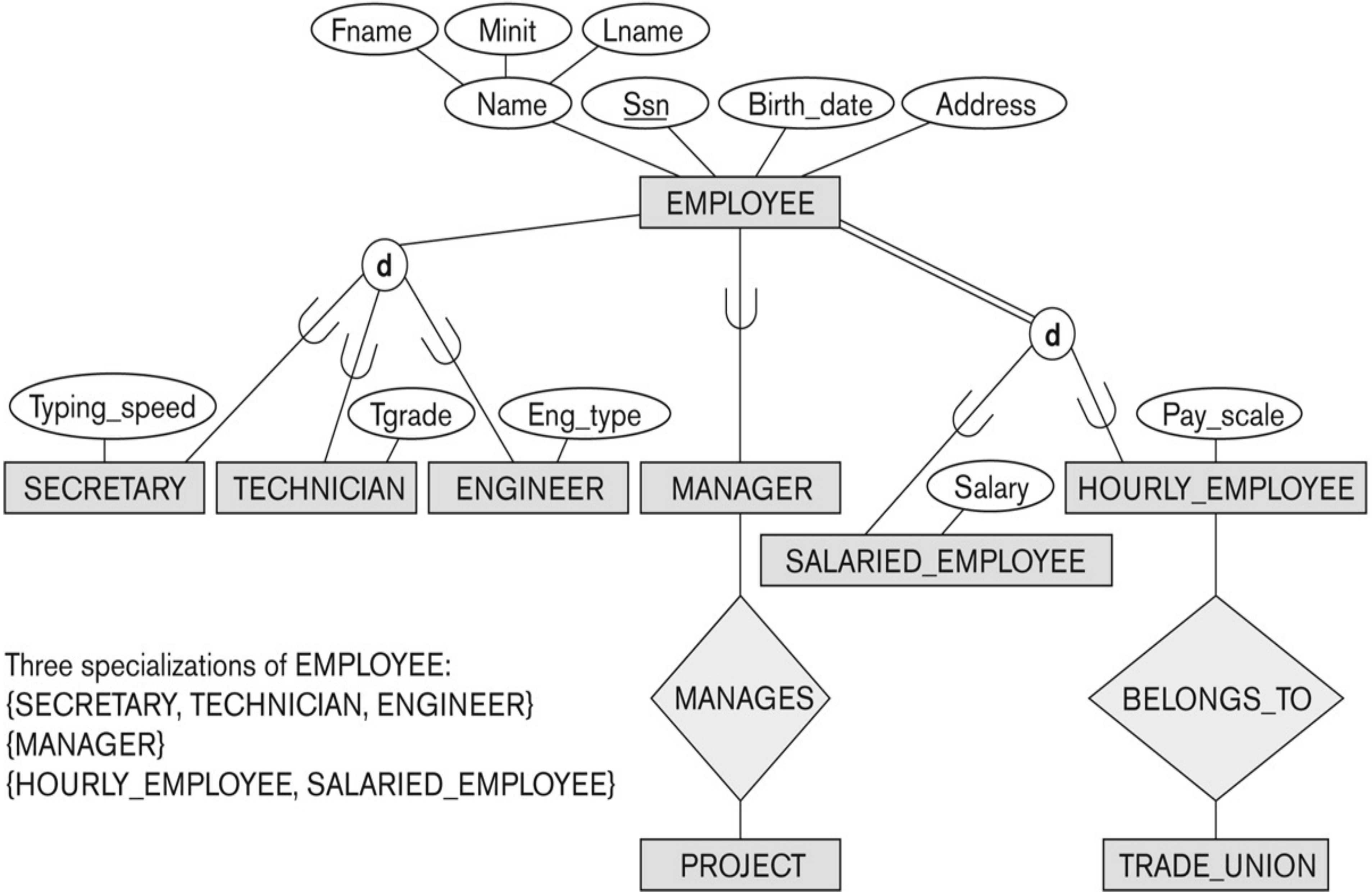
Enhanced-ER Model Concepts

- Includes all modeling concepts of basic ER
- Additional concepts: subclasses/superclasses, specialization/generalization, categories, attribute inheritance
- The resulting model is called the Enhanced-ER or Extended ER (E2R or EER) model
- It is used to model applications more completely and accurately if needed
- It includes some object-oriented concepts, such as inheritance

Subclasses and Superclasses

- An entity type may have additional meaningful subgroups of its entities
- Example: EMPLOYEE may be further grouped into SECRETARY, ENGINEER, MANAGER, TECHNICIAN, SALARIED_EMPLOYEE, HOURLY_EMPLOYEE,…
  - Each of these groups 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
  - Example: EMPLOYEE/SECRETARY, EMPLOYEE/TECHNICIAN

EER diagram notation to represent subclasses & specialization



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

Subclasses and Superclasses

- These are also called IS-A (IS-AN) relationships (SECRETARY IS-A EMPLOYEE, TECHNICIAN IS-A EMPLOYEE, …).
- Note: An entity that is a member of a subclass represents the same real-world entity as some member of the superclass
  - The Subclass member is the same entity in a distinct specific role
  - An entity cannot exist in the database merely by being a member of a subclass; it must also be a member of the superclass
  - A member of the superclass can be optionally included as a member of any number of its subclasses
    - →Example: A salaried employee who is also an engineer belongs to the two subclasses ENGINEER and SALARIED_EMPLOYEE
  - It is not necessary that every entity in a superclass be a member of some subclass
  - **Superclass/subclass relationship is one-to-one (1:1)**

Inheritance in Superclass/Subclass Relationships

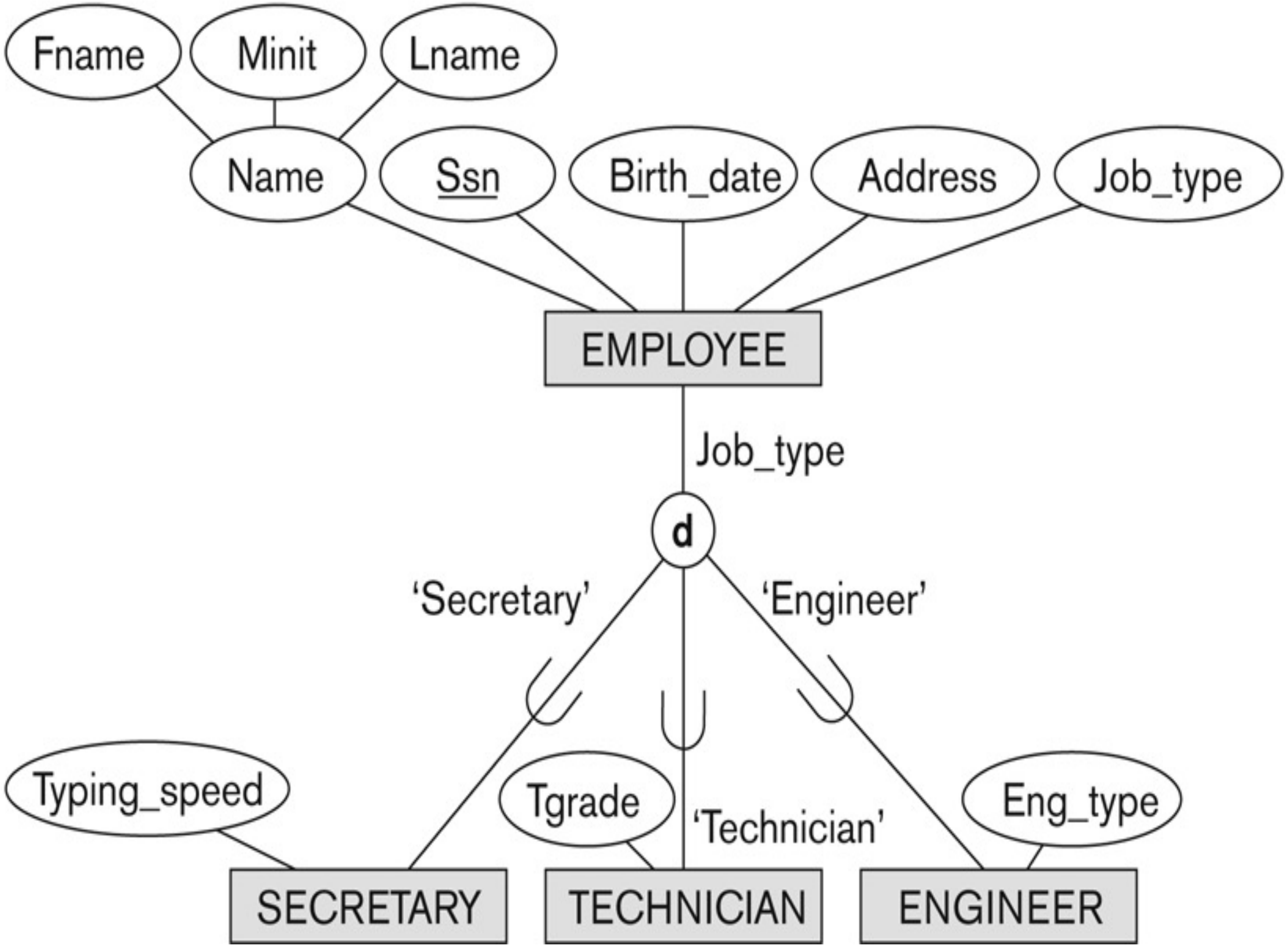
- An entity that is a member of a subclass *inherits* all attributes of the entity as a member of the superclass
- It also inherits all relationships

Specialization

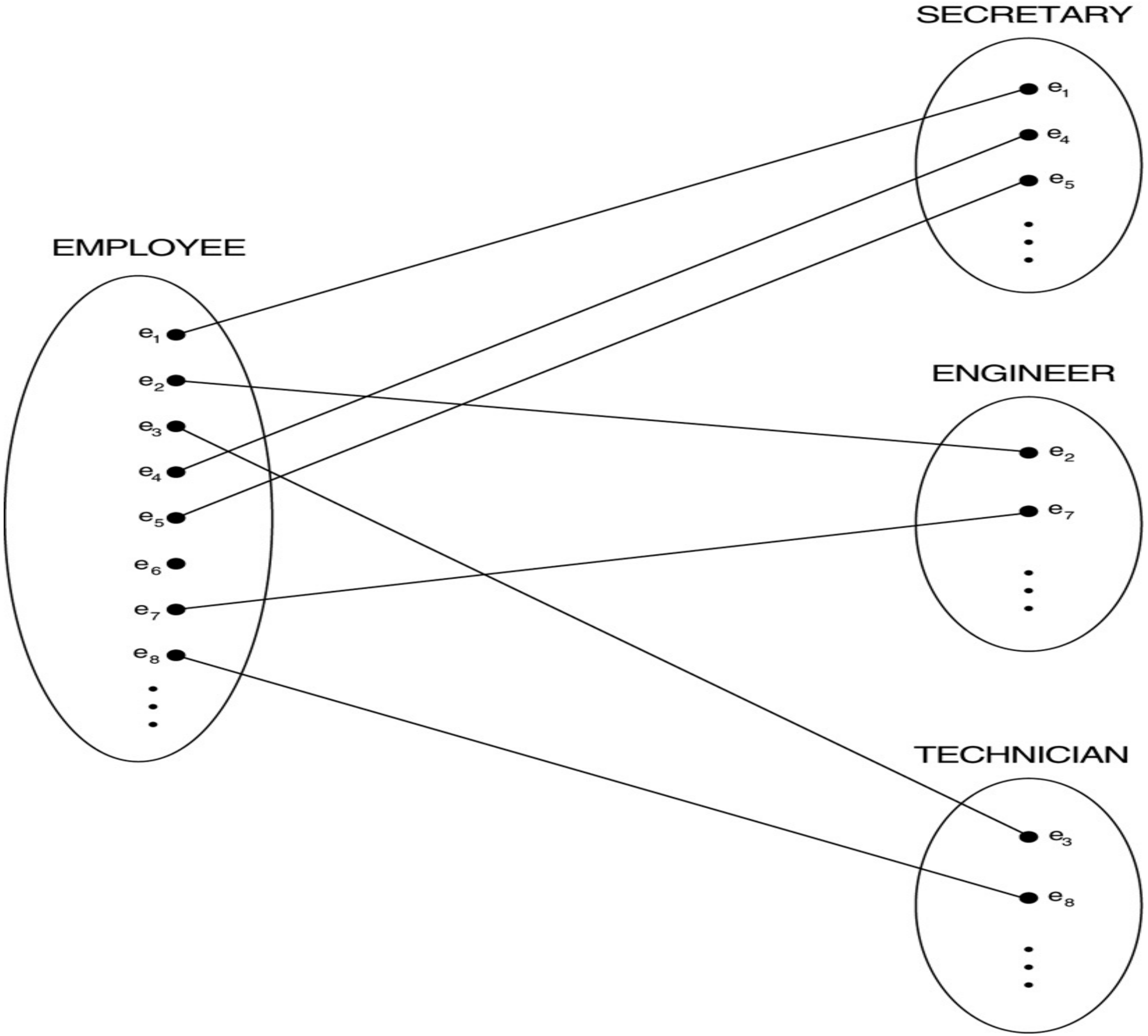
- Is the process of defining a set of subclasses of a superclass
- The set of subclasses is based upon some distinguishing characteristics of the entities in the superclass
- Example: {SECRETARY, ENGINEER, TECHNICIAN} is a specialization of EMPLOYEE based upon *job type*
  - May have several specializations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

Example of a Specialization

EER diagram notation for an attribute-defined specialization on Job_type.



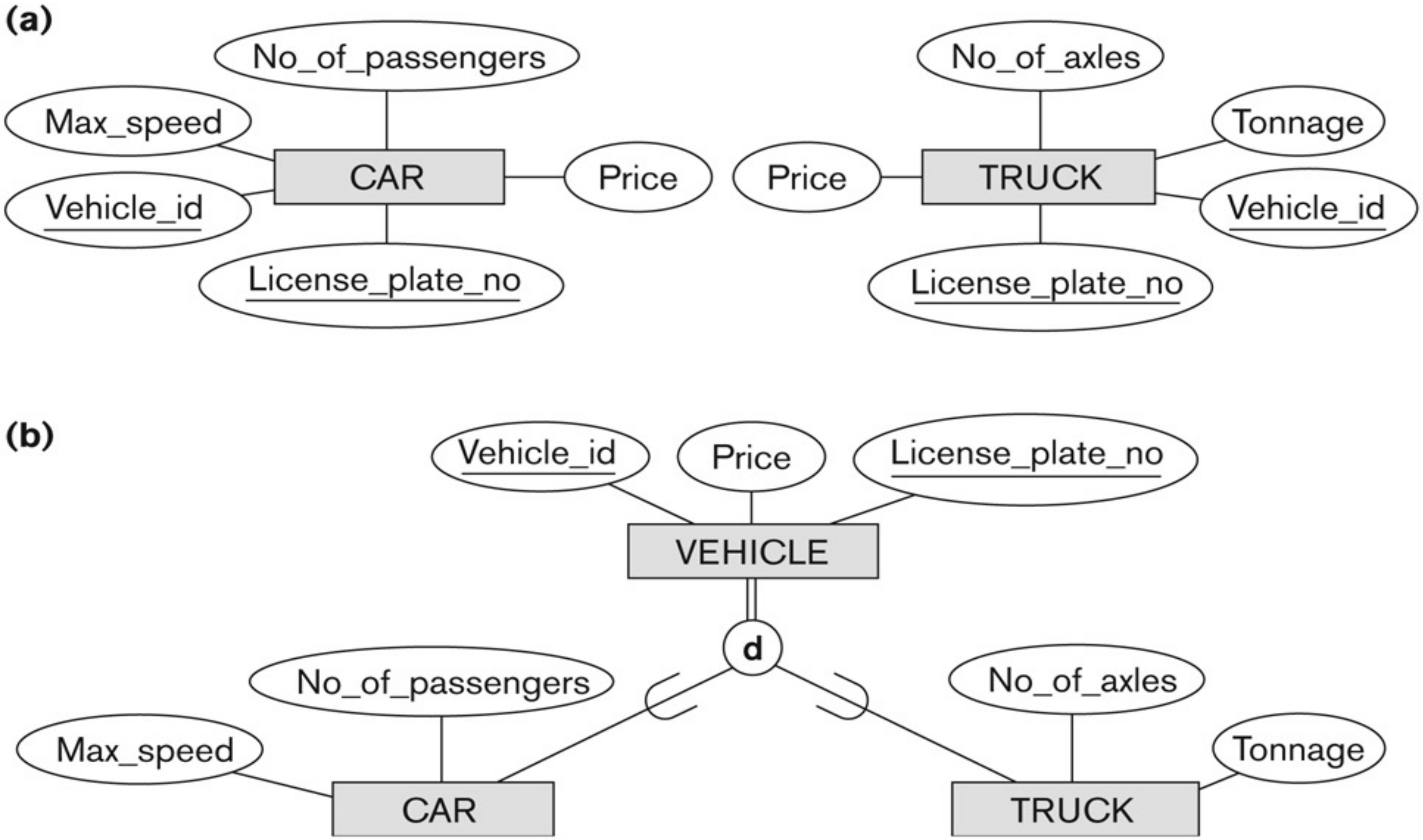
Instances of a specialization



Generalization

- 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 Example



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

Specialization and Generalization

- Diagrammatic notation sometimes used to distinguish between generalization and specialization
    - Arrow pointing to the generalized superclass represents a generalization
    - Arrows pointing to the specialized subclasses represent a specialization
    - We do not use this notation because it is often subjective as to which process is more appropriate for a particular situation
    - We advocate not drawing any arrows in these situations
- Data Modeling with Specialization and Generalization
    - A superclass or subclass represents a set of entities
    - Shown in rectangles in EER diagrams (as are entity types)
    - Sometimes, all entity sets are simply called classes, whether they are entity types, superclasses, or subclasses

Constraints on Specialization and Generalization

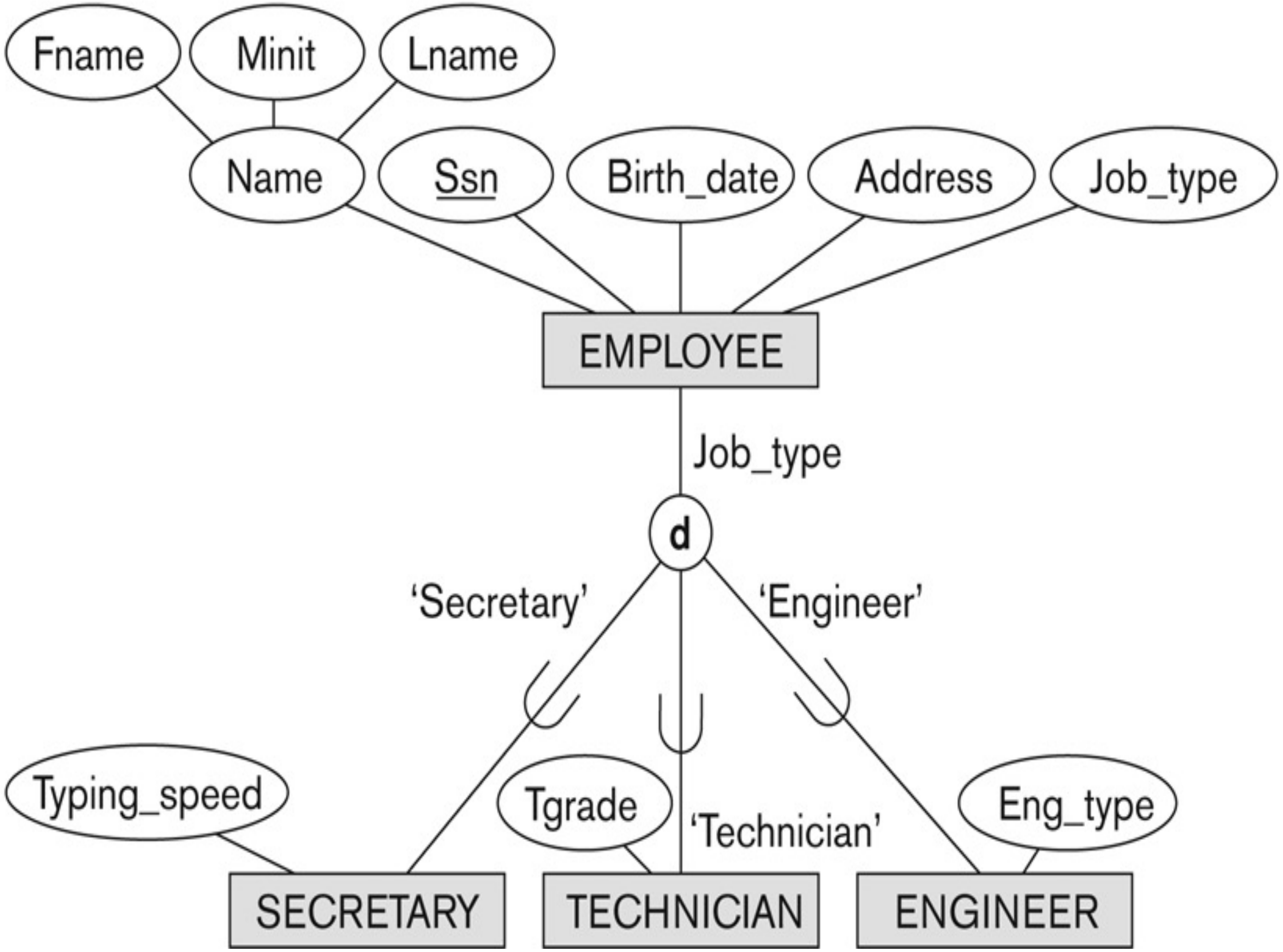
- If we can determine exactly those entities that will become members of each subclass by a condition, the subclasses are called *predicate-defined* (or *condition-defined*) subclasses
  - Condition is a constraint that determines subclass members
  - Display a predicate-defined subclass by writing the predicate condition next to the line attaching the subclass to its superclass

Constraints on Specialization and Generalization

- If all subclasses in a specialization have membership condition on same attribute of the superclass, specialization is called an *attribute defined*-specialization
    - Attribute is called the defining attribute of the specialization
    - Example: JobType is the defining attribute of the specialization {SECRETARY, TECHNICIAN, ENGINEER} of EMPLOYEE

EER diagram notation for an attribute-defined specialization on JobType

EER diagram notation for an attribute-defined specialization on Job_type.



Constraints on Specialization and Generalization

- If no condition determines membership, the subclass is called *user-defined*
  - Membership in a subclass is determined by the database users by applying an operation to add an entity to the subclass
  - Membership in the subclass is specified individually for each entity in the superclass by the user

Constraints on Specialization and Generalization

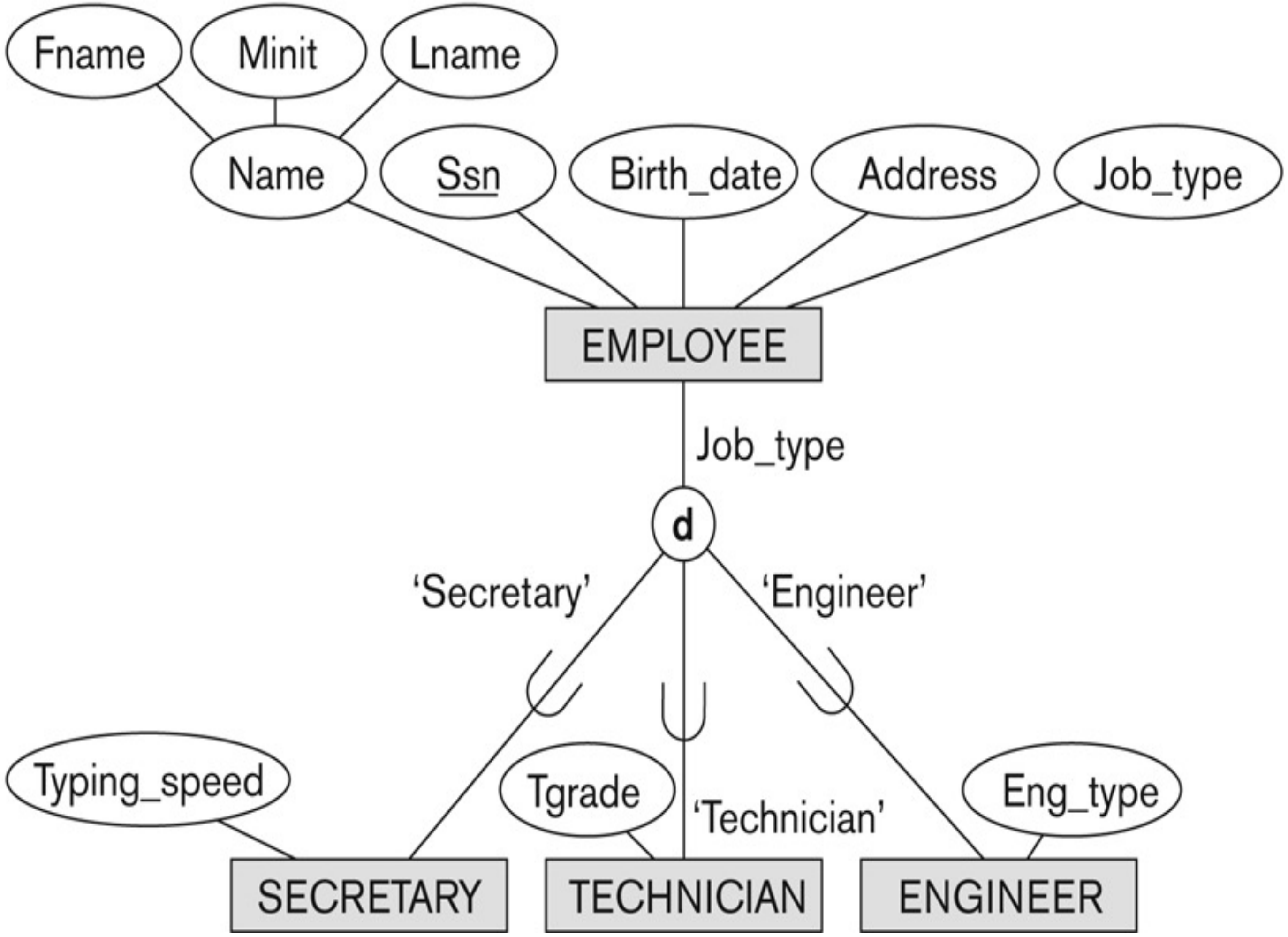
- Two basic conditions apply to a specialization or generalization: disjointness and completeness constraints
- **Disjointness Constraint**:
  - Specifies that the subclasses of the specialization must be disjointed (an entity can be a member of at most one of the subclasses of the specialization)
  - Specified by d in EER diagram
  - If not disjointed, overlap; that is the same entity may be a member of more than one subclass of the specialization
  - Specified by o in EER diagram

Constraints on Specialization and Generalization

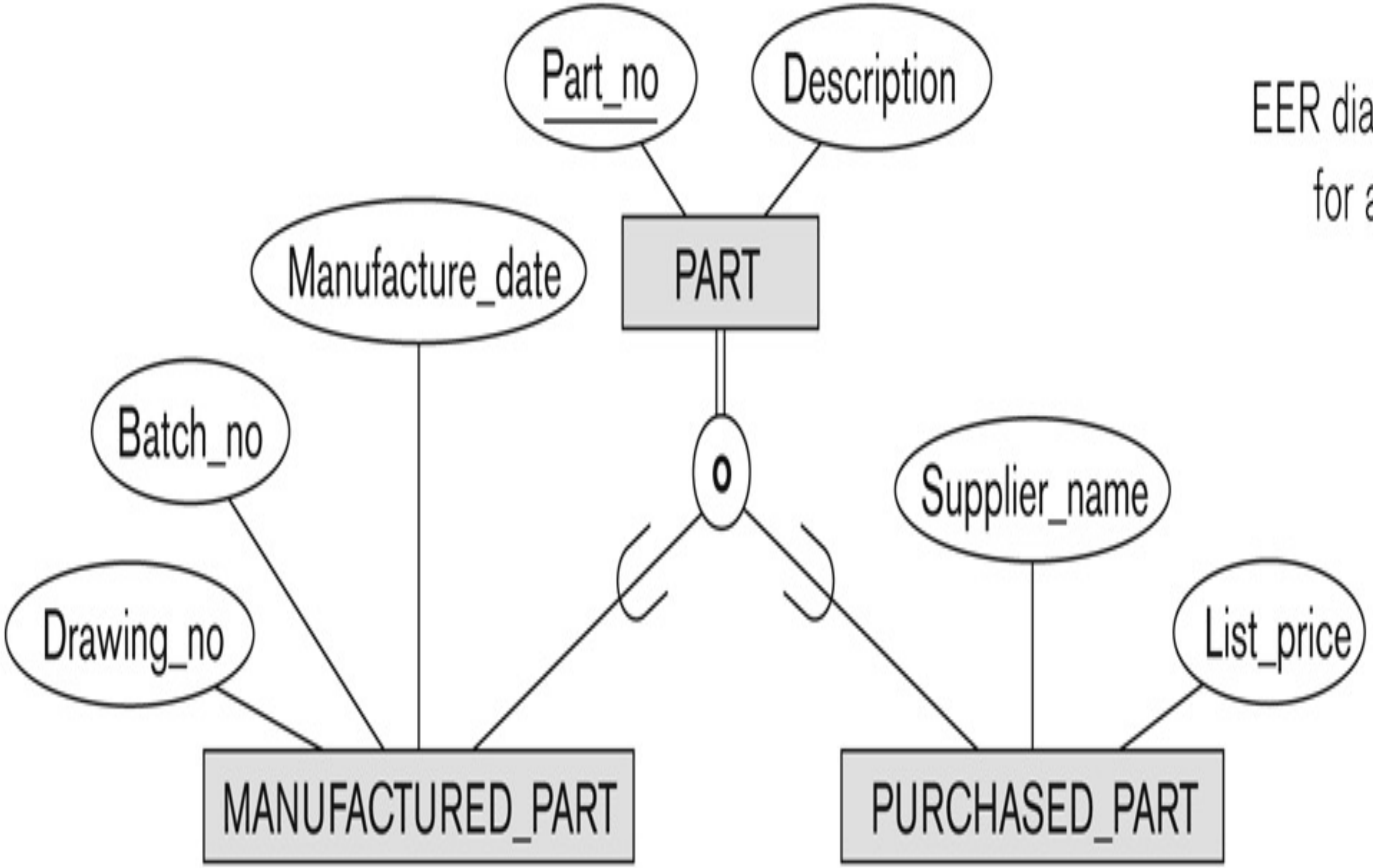
- **Completeness Constraint**:
  - Total specifies that every entity in the superclass must be a member of some subclass in the specialization/ generalization: Shown in EER diagrams by a double line
  - Partial allows an entity not to belong to any of the subclasses: Shown in EER diagrams by a single line

Example of Disjoint Partial Specialization

EER diagram notation for an attribute-defined specialization on Job_type.



Example of Overlapping Total Specialization



EER diagram notation
for an overlapping
(nondisjoint)
specialization.

Constraints on Specialization and Generalization

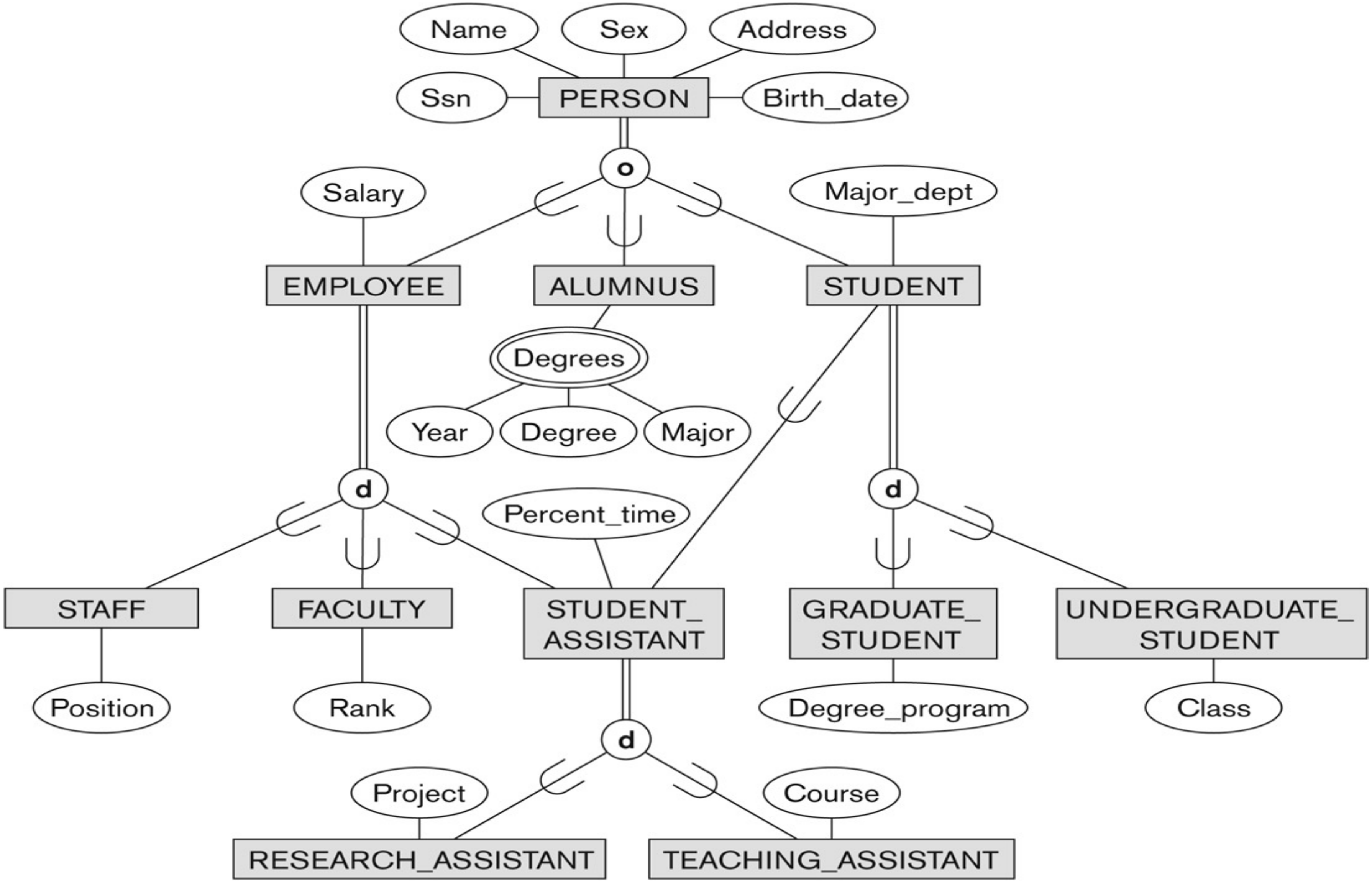
- Hence, we have four types of specialization / generalization:
  - Disjoint, total
  - Disjoint, partial
  - Overlapping, total
  - Overlapping, partial

- Note: Generalization is usually total because the superclass is derived from the subclasses

Specialization / Generalization Hierarchies, Lattices and Shared Subclasses

- A subclass may itself have further subclasses specified on it, forming a hierarchy or a lattice
- Hierarchy has a constraint that every subclass has only one superclass (called *single inheritance*)
- In a lattice, a subclass can be subclass of more than one superclass (called *multiple inheritance*)
- In a lattice or hierarchy, a subclass inherits attributes not only of its direct superclass, but also of all its predecessor superclasses
- A subclass with more than one superclass is called a shared subclass
- Can have specialization hierarchies or lattices, or generalization hierarchies or lattices

Specialization / Generalization Lattice Example (UNIVERSITY)



A specialization lattice with multiple inheritance for a UNIVERSITY database.

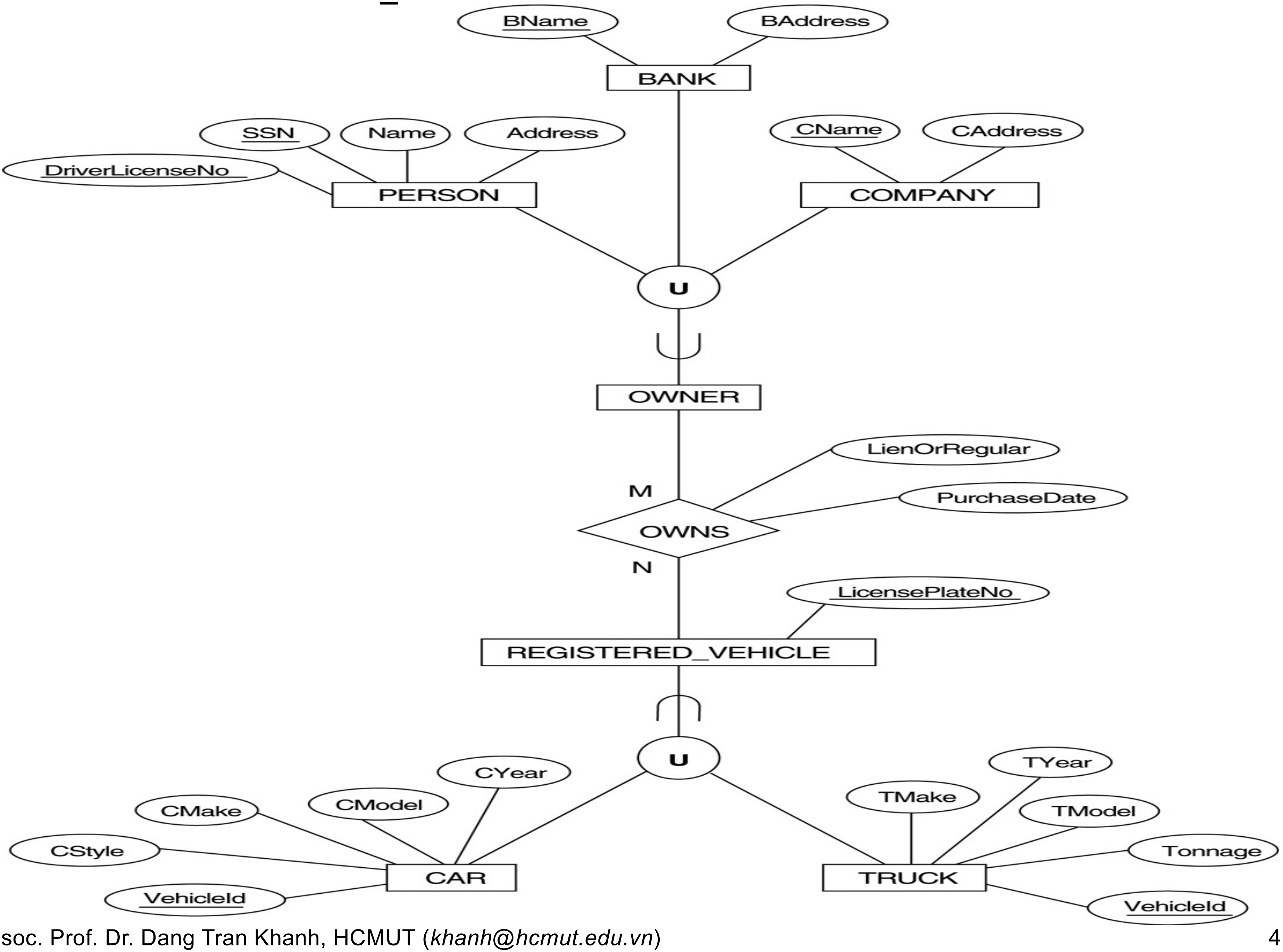
Categories

- All of the superclass/subclass relationships we have seen thus far have a single superclass
- A shared subclass is subclass in more than one distinct superclass/subclass relationships, where each relationship has a single superclass (multiple inheritance)
- In some cases, need to model a single superclass/subclass relationship with more than one superclass
- Superclasses represent different entity types
- Such a subclass is called a category or UNION TYPE

Categories

- Example: Database for vehicle registration, vehicle owner can be a person, a bank (holding a lien on a vehicle) or a company.
  - Category (subclass) OWNER is a subset of the union of the three superclasses COMPANY, BANK, and PERSON
  - A category member must exist in at least one of its superclasses
- Note: The difference from shared subclass, which is a subset of the intersection of its superclasses (shared subclass member must exist in all of its superclasses)

Two categories (union types): OWNER and REGISTERED_VEHICLE



Formal Definitions of EER Model

- Class C:
  - A type of entity with a corresponding set of entities:
    - →could be entity type, subclass, superclass, or category
- Note: The definition of *relationship type* in ER/EER should have 'entity type' replaced with 'class‘ to allow relationships among classes in general
- Subclass S is a class whose:
    - →Type inherits all the attributes and relationship of a class C
    - →Set of entities must always be a subset of the set of entities of the other class C: $S \subseteq C$
    - →C is called the superclass of S
    - →A superclass/subclass relationship exists between S and C

Formal Definitions of EER Model

- Specialization Z: Z = {S1, S2,…, Sn} is a set of subclasses with same superclass G; hence, G/Si is a superclass/subclass relationship for i = 1,…, n
  - G is called a generalization of the subclasses {S1, S2,…, Sn}
  - Z is total if we always have:
    - →S1 ∪ S2 ∪ … ∪ Sn = G;
    - →Otherwise, Z is partial
  - Z is disjoint if we always have:
    - →Si ∩ Sj empty-set for i ≠ j;
    - →Otherwise, Z is overlapping

Formal Definitions of EER Model

- Subclass S of C is predicate defined if predicate (condition) p on attributes of C is used to specify membership in S; that is, S = C[p], where C[p] is the set of entities in C that satisfy condition p
- A subclass not defined by a predicate is called user-defined
- Attribute-defined specialization: if a predicate A = ci (where A is an attribute of G and ci is a constant value from the domain of A) is used to specify membership in each subclass Si in Z
  - Note: If ci ≠ cj for i ≠ j, and A is single-valued, then the attribute-defined specialization will be disjoint.

Formal Definitions of EER Model

- Category or UNION type T
  - A class that is a subset of the *union* of n defining superclasses
    D1, D2,…Dn, n>1:
    - $\rightarrow T \subseteq (D1 \cup D2 \cup \ldots \cup Dn)$
  - Can have a predicate pi on the attributes of Di to specify entities of Di that are members of T.
  - If a predicate is specified on every Di: $T = (D1[p1] \cup D2[p2] \cup \ldots \cup Dn[pn])$

EER model

- Limitations of Basic Concepts of the ER Model
- Enhanced-ER (EER) Model Concepts
- Subclasses and Superclasses
- Specialization and Generalization
- Specialization / Generalization Hierarchies, Lattices and Shared Subclasses
- Categories
- Formal Definitions of EER Model
- Database Design Modeling Tools

Database Design Modeling Tools: Examples

COMPANY	TOOL	FUNCTIONALITY
Embarcadero Technologies	ER Studio	Database Modeling in ER and IDEF1X
	DB Artisan	Database administration and space and security management
Oracle	Developer 2000 and Designer 2000	Database modeling, application development
Popkin Software	System Architect 2001	Data modeling, object modeling, process modeling, structured analysis/design
Platinum Technology (Computer Associates)	Platinum Enterprice Modeling Suite: Erwin, BPWin, Paradigm Plus	Data, process, and business component modeling
Persistence Inc.	Pwertier	Mapping from O-O to relational model
Rational (IBM)	Rational Rose	Modeling in UML and application generation in C++ and JAVA
Rogue Ware	RW Metro	Mapping from O-O to relational model
Resolution Ltd.	Xcase	Conceptual modeling up to code maintenance
Sybase	Enterprise Application Suite	Data modeling, business logic modeling
Visio (Microsoft)	Visio Enterprise	Data modeling, design and reengineering Visual Basic and Visual C++
…		

Outline

- ERD exercises
- Introduction to ER model weaknesses
- Enhanced/Extended ER model
- Reading:
  - [1]: Chapter 4 (7th ed)

Summary

- ERD exercises
- Introduction to ER model weaknesses
- Introduction to EER model
  - Limitations of Basic Concepts of the ER Model
  - Enhanced-ER (EER) Model Concepts
  - Subclasses and Superclasses
  - Specialization and Generalization
  - Specialization / Generalization Hierarchies, Lattices and Shared Subclasses
  - Categories
  - Formal Definitions of EER Model
  - Database Design Modeling Tools

Q&A

Question ?