

DATABASE ANALYSIS AND DESIGN 3

Week 5

Outline

- Enhanced Entity-Relationship Modelling
 - Superclass / subclass
 - Specialization and Generalization
 - Categorisation

Enhanced ER (EER) Modelling

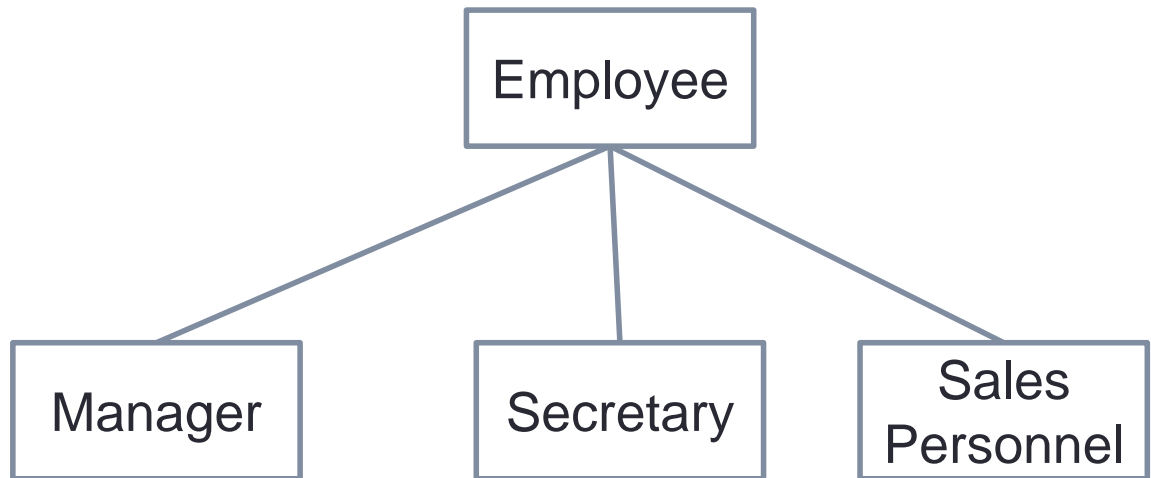
- Since the 1980s there has been an increase in the emergence of new database applications with more demanding requirements.
- Basic concepts of ER modelling are not sufficient to represent the requirements of the newer, more complex applications.
- Response is development of additional 'semantic' modelling concepts.
- Semantic concepts are incorporated into the original ER model and is called the Enhanced Entity-Relationship (EER) model.
 - That is EER includes all modelling concepts in ER

EER Modelling

- EER is also Known as Extended ER model
- Created to design more accurate database schemas
- Reflect the data properties and constraints more precisely
- Additional concepts of EER model includes specialization / generalization, and categorization.

Superclass and Subclass

- A **Superclass** – is an entity type that includes distinct Subclasses that require to be represented in a data model.
- A **Subclass** is an entity type that has a distinct role and is also a member of the Superclass

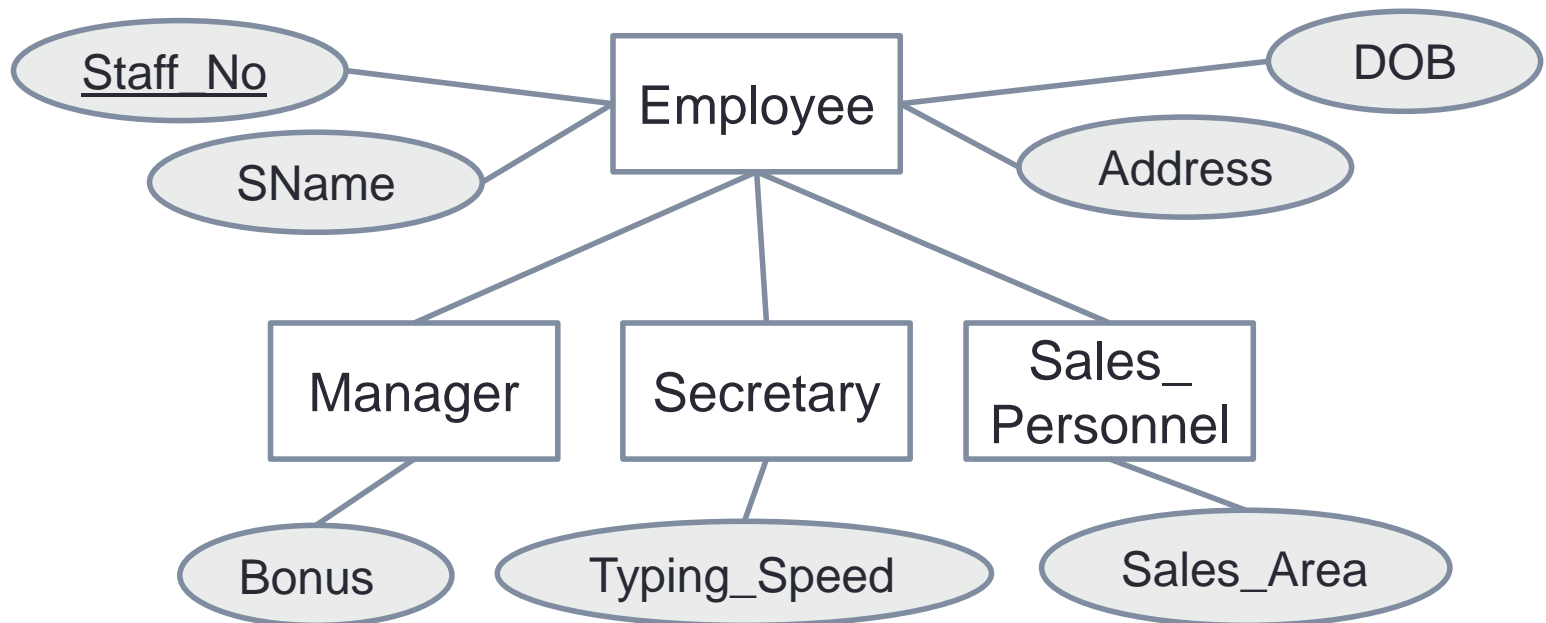


Superclass and Subclass Examples

- Member (superclass): Life_Member, Regular_Member, Season_Member (subclasses)
- Staff (superclass): Full_Time_Staff, Part_Time_Staff (subclasses)
- It adds more semantic clarity
 - If only Full_Time_Staff belongs to the Welfare_Society then it can be represented as
 - *Belongs_To* : <Full_Time_Staff, Welfare_Society>
 - **Not as** *Belongs_To* : <Staff, Welfare_Society>

Inheritance

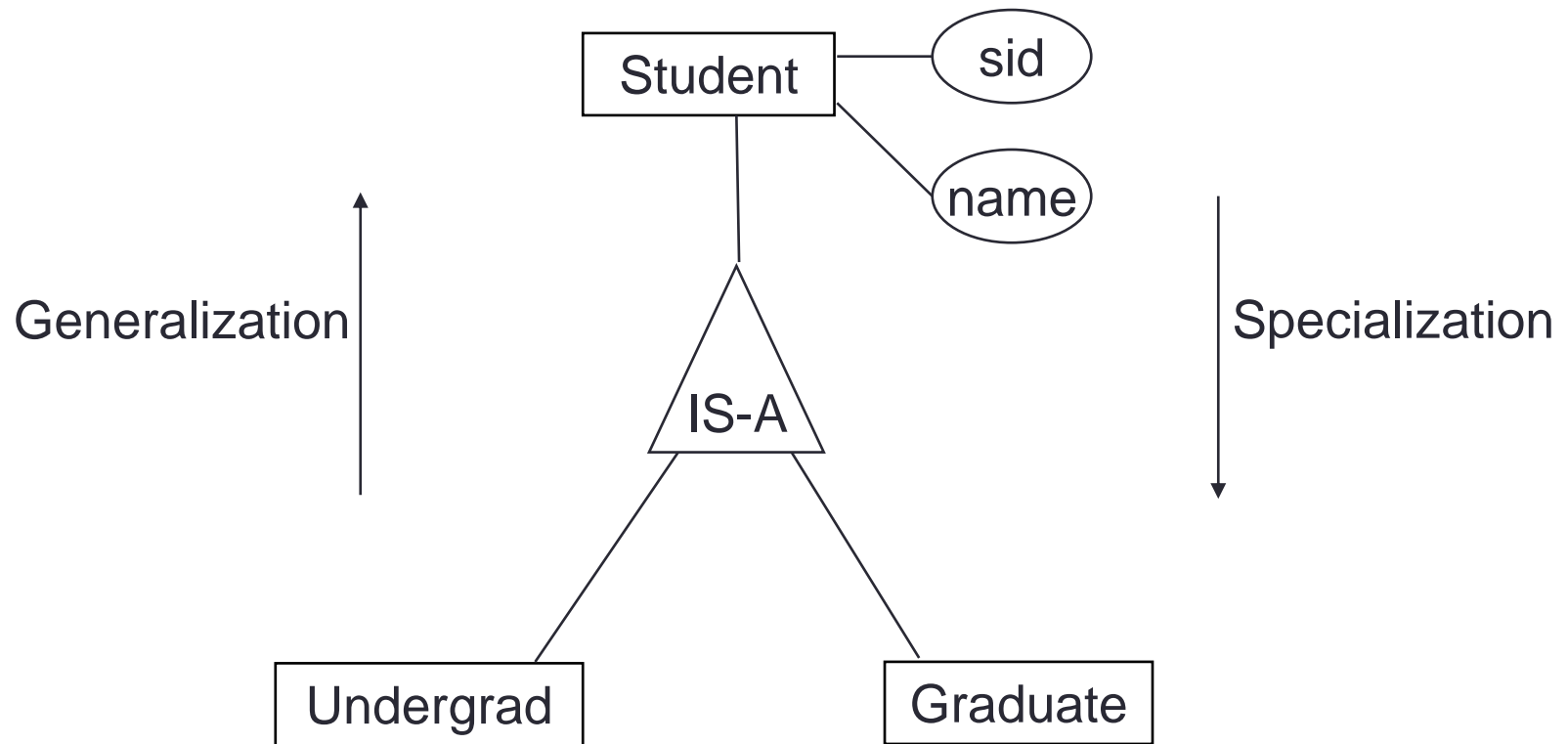
- Subclass entity inherits all attributes and relationships of superclass
- **Attribute Inheritance**
 - An entity in a Subclass may possess subclass specific attributes, as well as those associated with the Superclass.



IS-A ('is a') Hierarchies

- Inheritance IS-A (instance) Relationship
- If we declare A **is a** B. Every A entity is considered to be a B entity.
- Secretary **IS-A** Employee
- Single inheritance results in a hierarchy (tree structure)
 - Every subclass has only one superclass
- Multiple inheritances results in a lattice
 - Subclass with more than one superclass
 - E.g. Student → Student_Assistant <- Employee

Is-A Relationship



Specialization and Generalization

- **Specialization**

- The process of maximizing the differences between members of an entity by identifying their distinguishing characteristics.
- Identify subclasses and their *distinguishing characteristics* (attributes, relationships)
- Start with entity type then define subclasses by successive specialization



Top-Down

- **Generalization**

- The process of minimizing the differences between entities by identifying their common characteristics.
- Aggregates entities into a superclass entity types *by identifying their common characteristics*



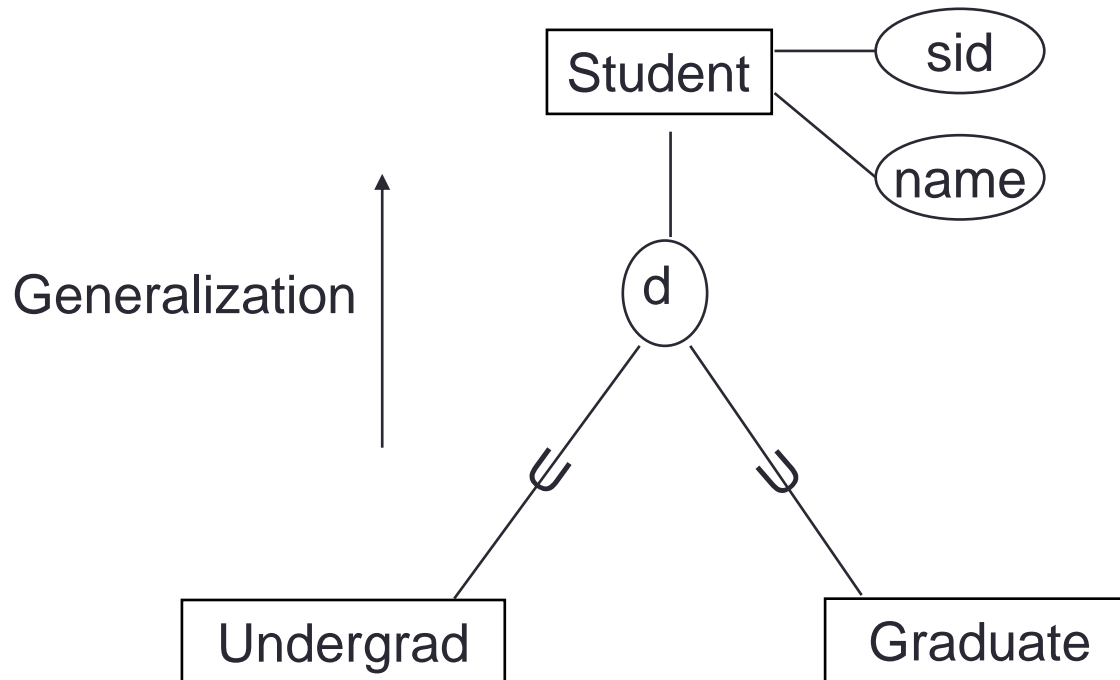
Bottom up

Specialization and Generalization

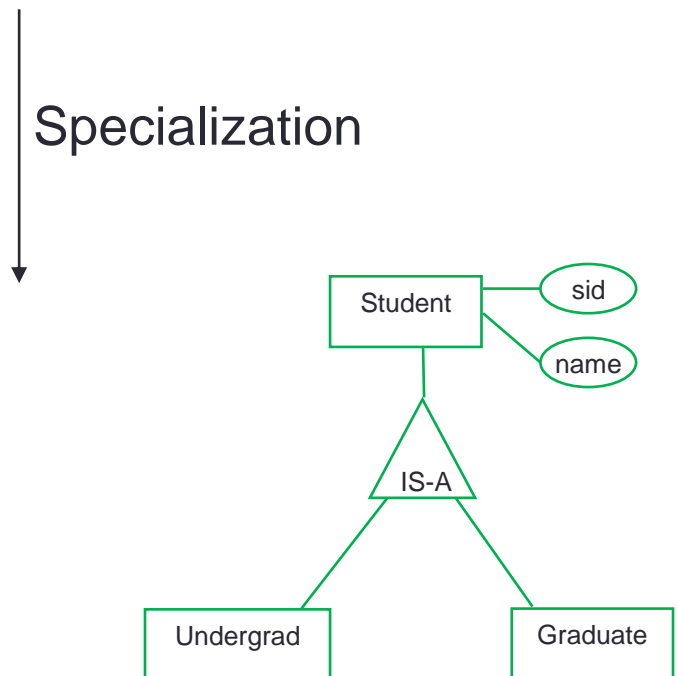
- CAR, TRUCK generalized into VEHICLE;
- Both CAR, TRUCK become subclasses of the superclass VEHICLE.
 - We can view {CAR, TRUCK} as a specialization of VEHICLE
 - Alternatively, we can view VEHICLE as a generalization of CAR and TRUCK

Specialization and Generalization

An alternative notation is the Union symbol



The **circle with d** specifies that the specializations are disjoint. A member of Undergrad entity set NOT a member of the graduate entity set.



Reasons for Specialization?

- Certain attributes may apply to some but not all entities of the superclass
 - E.g. Secretary – typing speed
 - Thus, it minimizes Null Values
- Some relationship types may be participated in only by entities that are members of the subclass
 - If only Full_Time Employee belongs to Welfare_Society

Specific (local)
attributes of
subclass

Specific
relationship

Specialization and Generalization

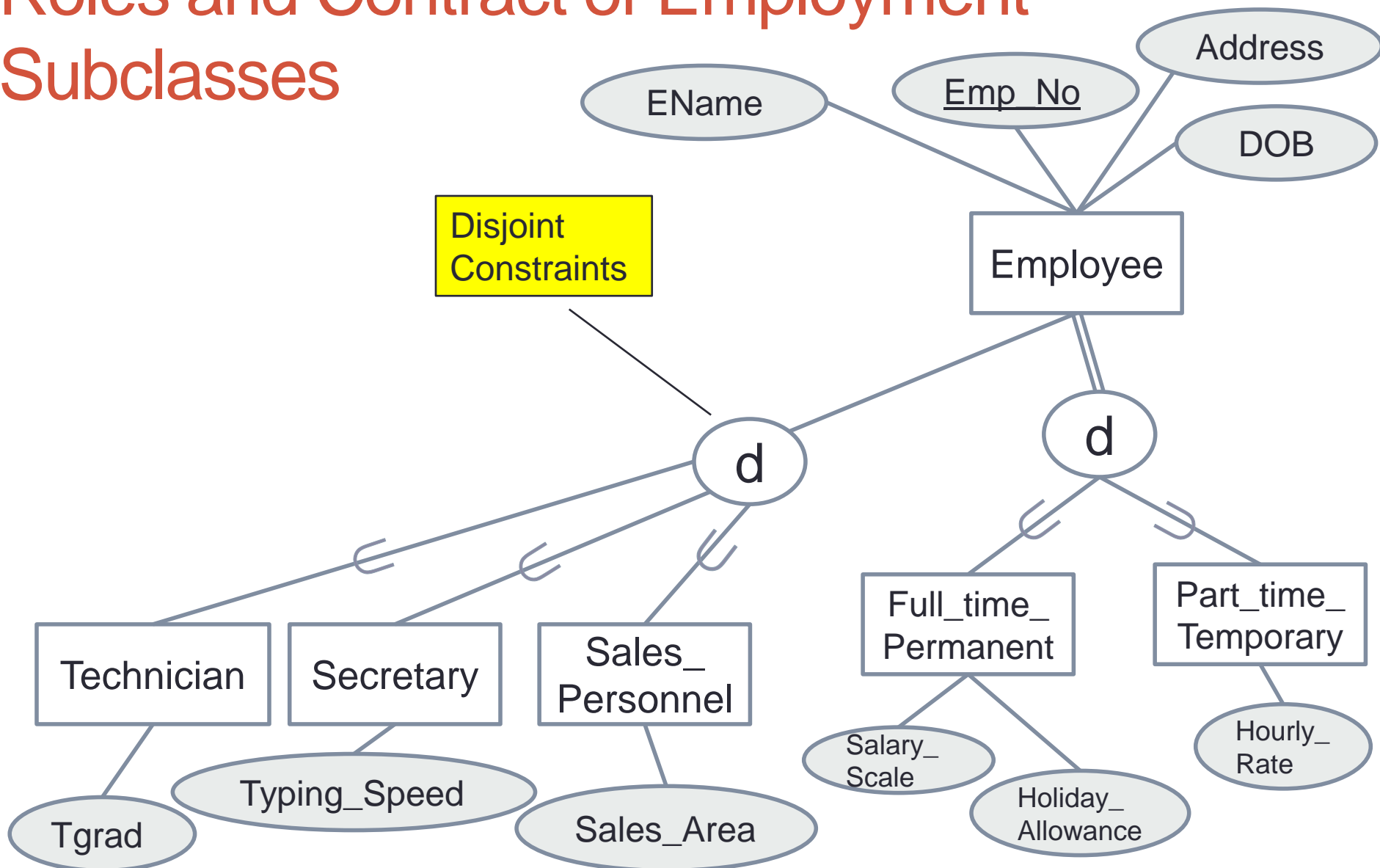
Specialization and generalization has:

- Disjointness (Inclusion) Constraints
- Completeness (Participation) Constraints
- Disjointness and Completeness constraints are independent

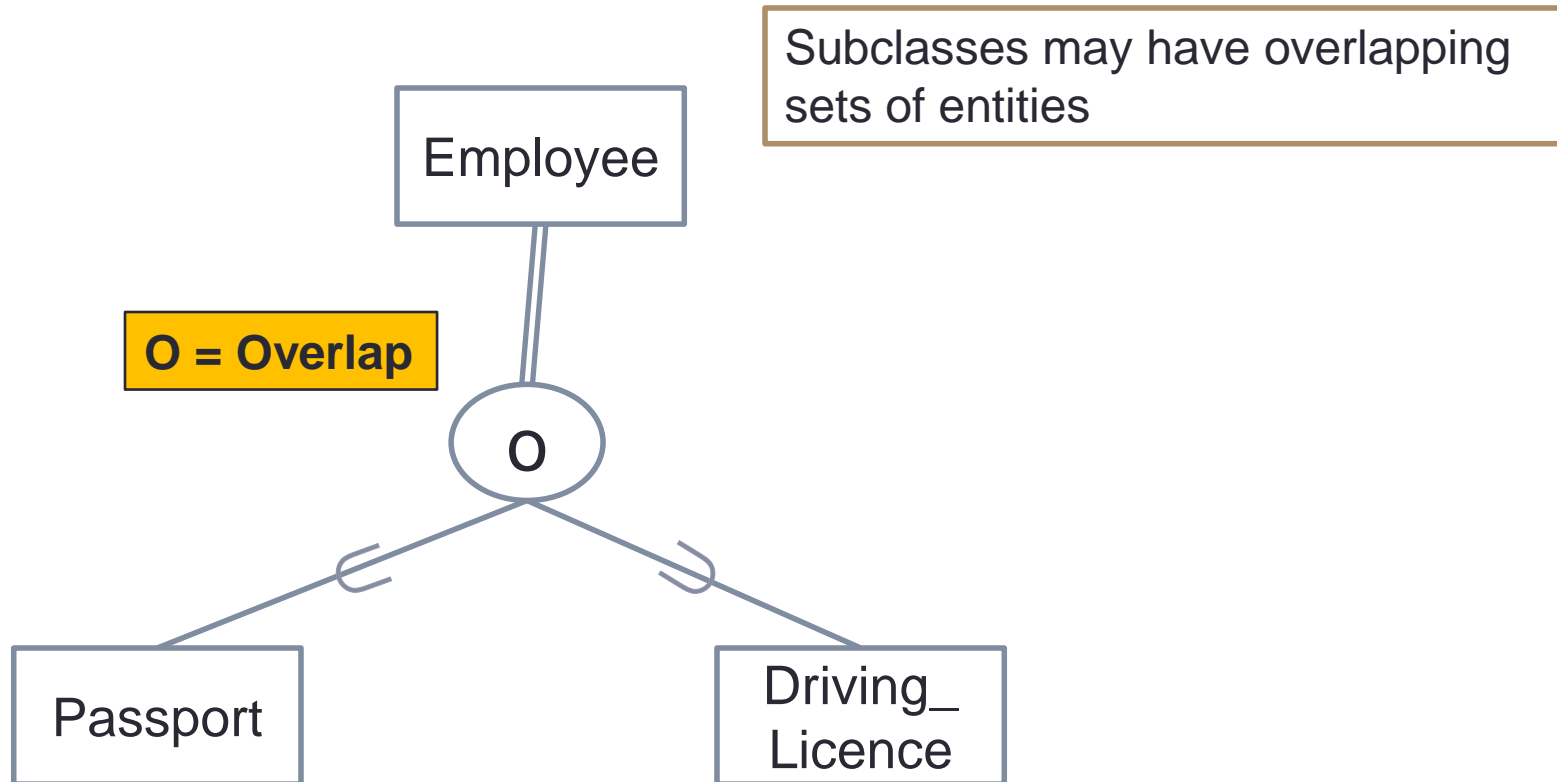
Disjointness Constraint

- Disjoint
 - Members in different Subclasses from the same Superclass are completely different.
 - An entity can be a member of at most one of the subclasses of the specialization
 - Specified by **d** in EER diagram
- Non-Disjoint (overlap)
 - Members in a Superclass can be enrolled in more than one Subclass
 - That is the same entity may be a member of more than one subclass of the specialization
 - Specified by **o** in EER diagram

Specialization of Employee Entity into Job Roles and Contract of Employment Subclasses



Non-Disjoint Constraints



Completeness Constraints

- **Total specialization**

- All member in the Superclass must participate in either one Subclass.
- Shown in EER diagrams by a double line

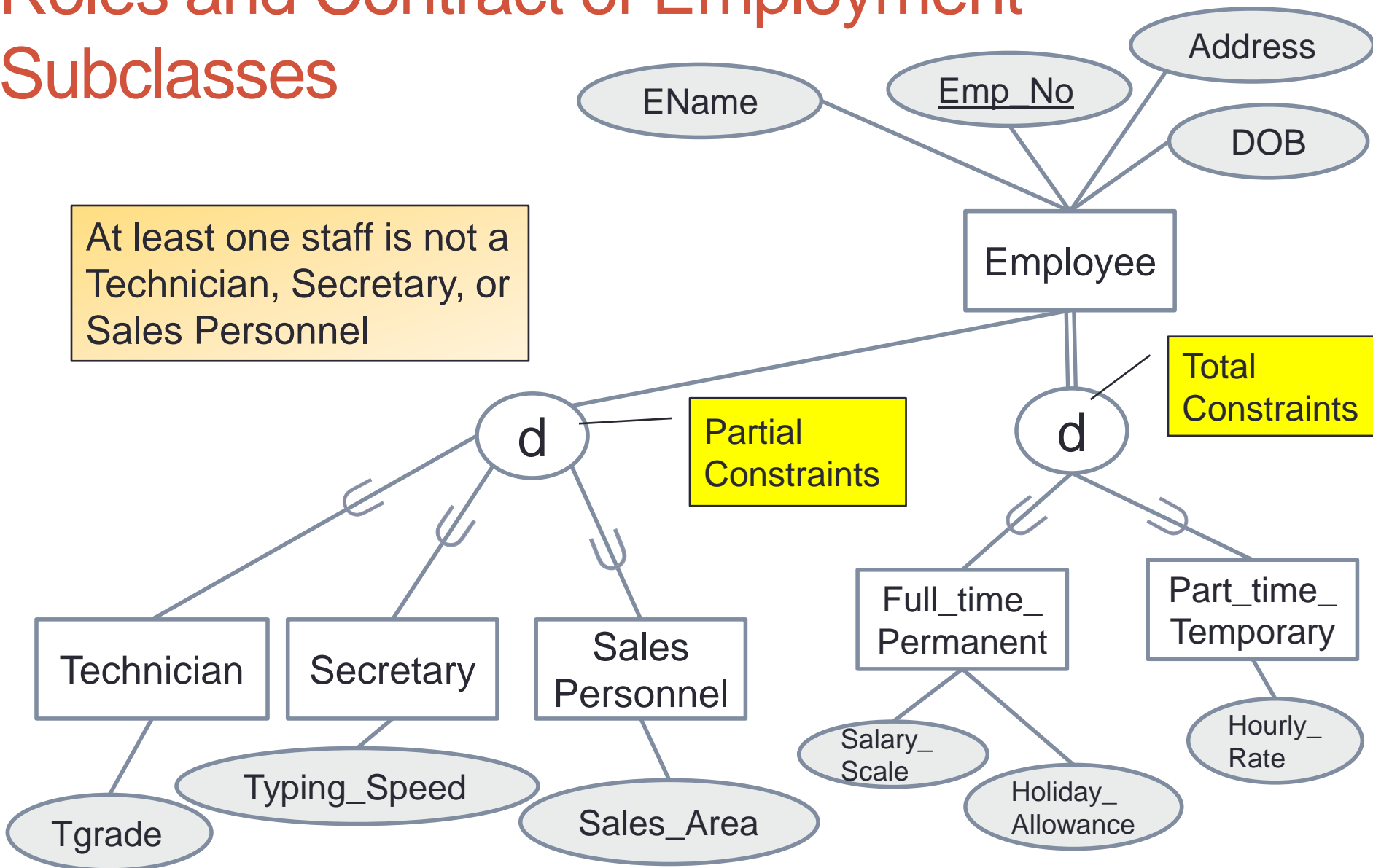
- **Partial specialization**

- At least one member in the Superclass does not participate in the Subclass.
- Partial allows an entity not to belong to any of the subclasses
- Shown in EER diagrams by a single line

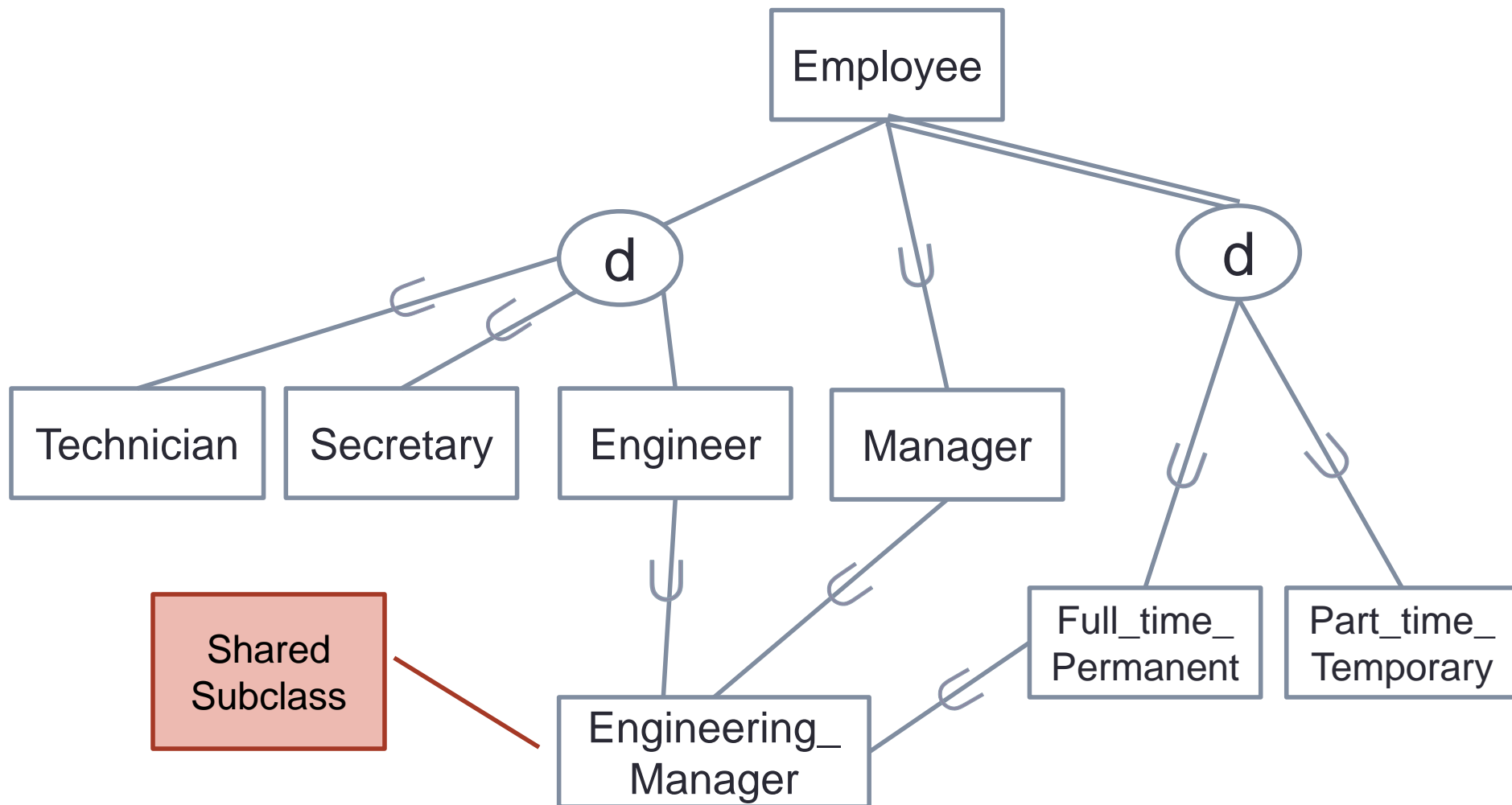
- Generalization usually is total because the superclass is derived from the subclasses.

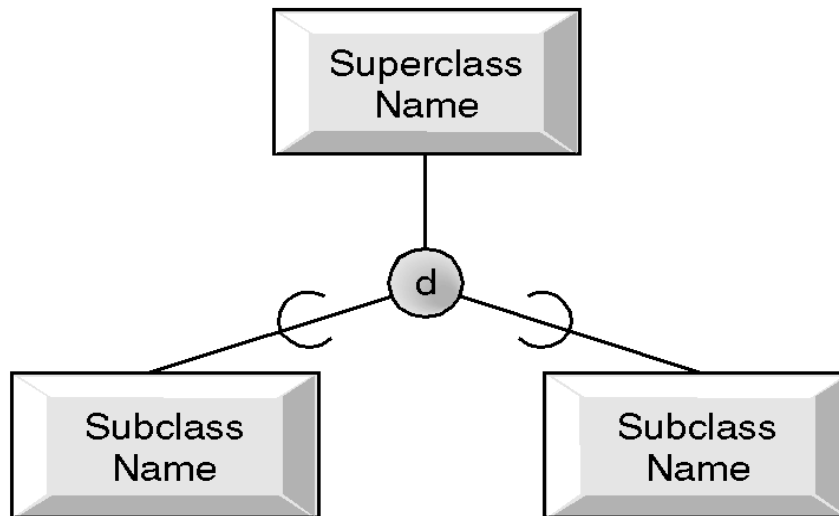
Specialization of Employee Entity into Job Roles and Contract of Employment Subclasses

At least one staff is not a Technician, Secretary, or Sales Personnel

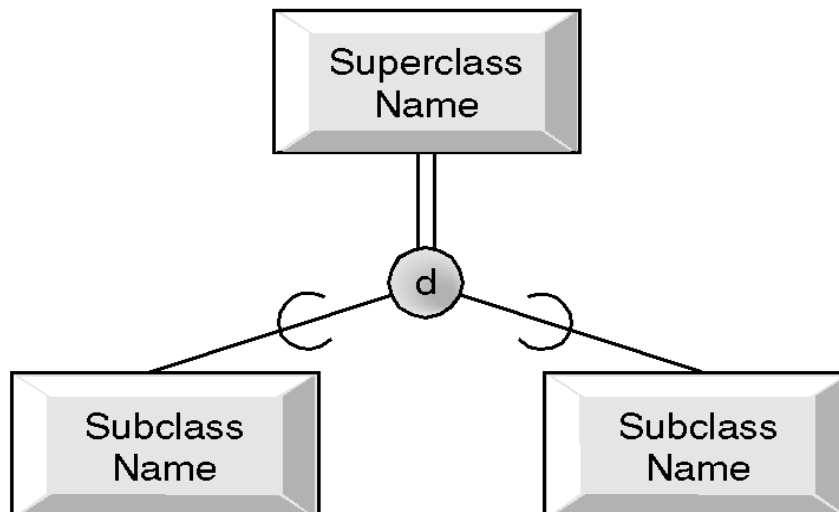


Specialization Lattice (Multiple Inheritance)



Notation**Meaning**

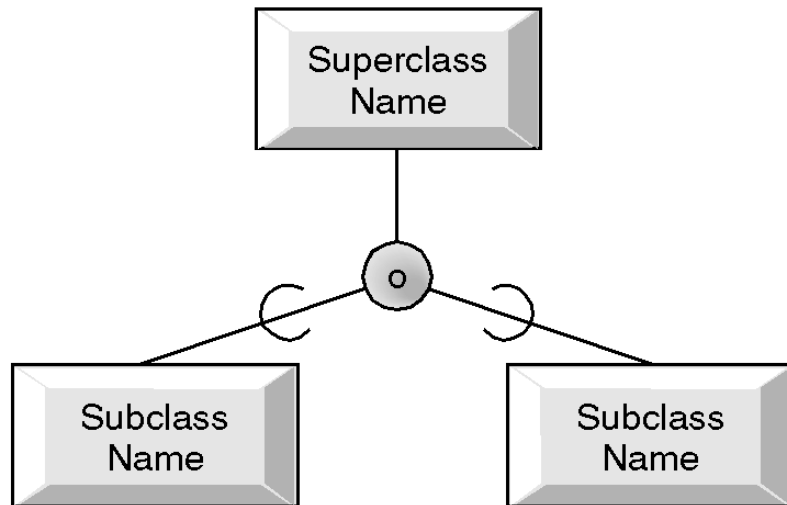
Superclass/Subclass Relationship
Disjoint, Partial



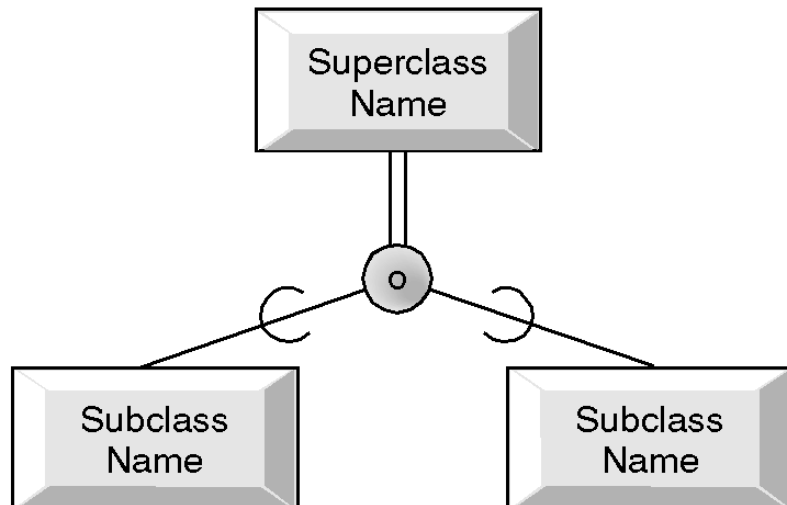
Superclass/Subclass Relationship
Disjoint, Total

Notation

Meaning



Superclass/Subclass Relationship
Overlapping, Partial



Superclass/Subclass Relationship
Overlapping, Total

Insertion and deletion rules applying to specialization and generalization

- Deleting an entity from a superclass implies that it is automatically deleted from all the subclasses to which belongs.
- Inserting an entity in a superclass implies that the entity is mandatorily inserted in all subclasses (in predicate defined or attribute defined subclasses).
- Inserting an entity in a superclass of a total specialization implies that the entity is mandatorily inserted in at least one of the subclasses of the specialization.

Building EER

- Identify entity types.
- Identify relationship types.
- Determine cardinality and participation constraints of relationship types.
- Identify and associate attributes with entity or relationship types.
- Determine candidate and primary key attributes.
- Specialize / generalize entity types.
- Categorize entity types.
- Draw the EER Diagram.