Enhanced Entity Relationship Model (EER)

The EER model includes all the modeling concepts of ER model plus the concept of subclass and superclass and the related concepts of specialization and generalization. Another concept included in the EER model is category or union type. In association with all these concepts inheritance relationship mechanism is also included in EER diagram.

Super class and Subclass

Super class is an ~~Entity~~ entity that can be devided into further subtypes.

Subclasses are the group of entities of a superclass with some unique characteristics.

Consider EMPLOYEE entity type. This entity type defines the type (i.e. attributes and relationships) of each EMPLOYEE entities and also refers to the current set of EMPLOYEE entities (i.e. the EMPLOYEE entity set).

Also, there are various subtypes of EMPLOYEE entity type which is meaningful and need to be represented.

for example the EMPLOYEE entity type may further subdevided into SECRETARY, ENGINEER, MANAGER, CLERK, ASSISTANTS etc.

Each subgrouping or subtype are known as subclass or subtype of the EMPLOYEE entity and the EMPLOYEE entity is known as super class or supertype of these subclass/subtypes.

The relationship between superclass and its subclasses are known as superclass/subclass relationship or supertype/subtype relationships or class/subclass relationships

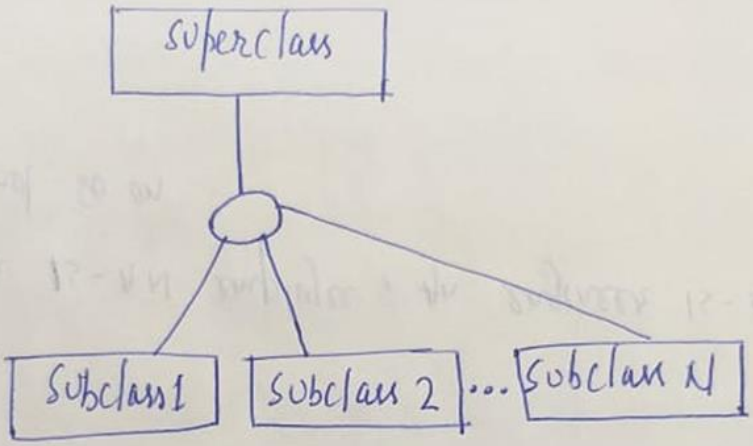
Note :→ Generic attributes are attached to the super class which is common to all the subclasses.

→ Relevant specific attributes are attached to the subclasses.

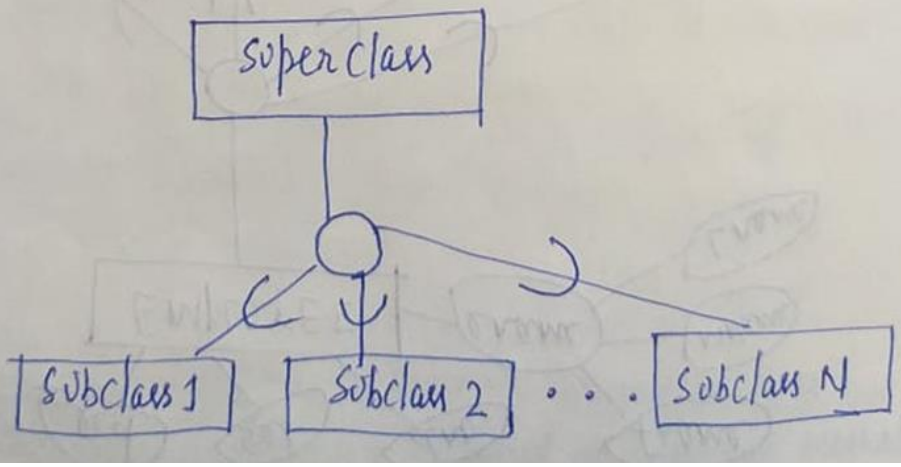
→ A superclass/subclass relationship is also known as IS-A (or IS-AN) relationship.

Notation

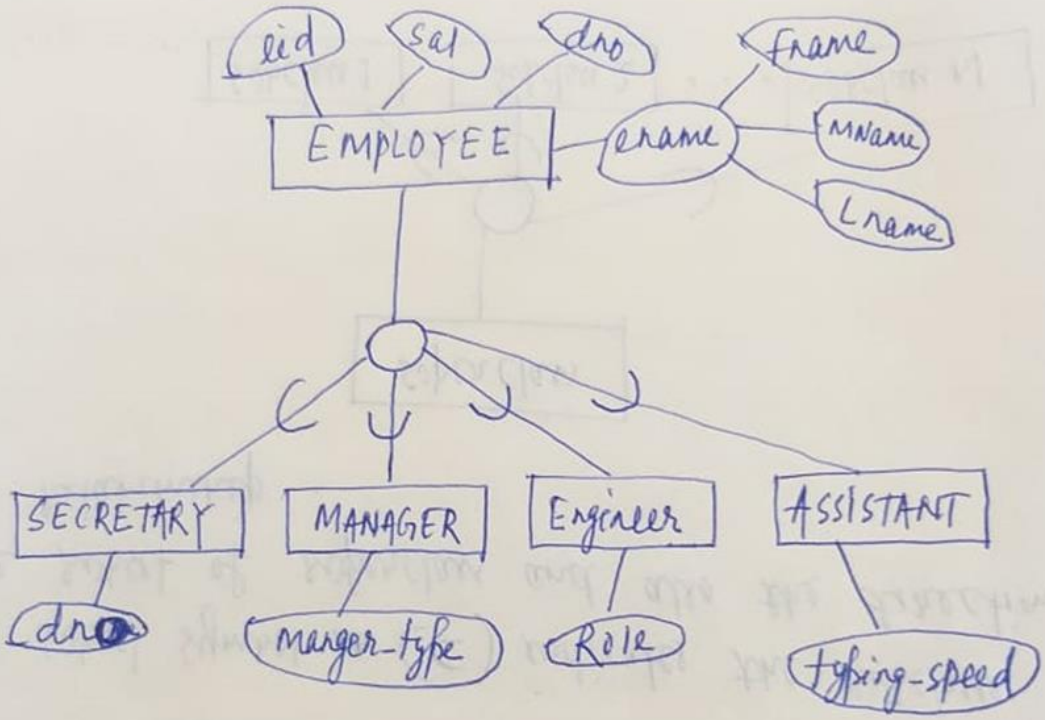
Subclasses are attached to superclass with a circle.



The subset symbol (⊂) indicates the subclasses are subset of superclass and also the direction of relationship.



e.g.



A manager IS-AN employee, An engineer IS-AN employee and so on.

Specialization

→ Specialization is a process of defining a set of subclass of an entity type i.e. a superclass.

→ It is a top-down approach where one entity (superclass) is devided into subclasses with some distingueishing characteristics of the entities in the super class.

for e.g. the subclasses {SECRETARY, ENGINEER, MANAGER} is a specialization of the super class EMPLOYEE that distingueishing among EMPLOYEE entities with respect to their designation.

There are several specialization of the same entity type is possible based on different distingueishing characteristics.

e.g. {SALARIED_EMPLOYEE, HOURLY_EMPLOYEE} is also a specialization of EMPLOYEE based on payment Methods.

GENERALIZATION

Generalization is the reverse process of specialization. In generalization several entity types with common features are grouped together and a generalized superclass is formed. The original entity types of common features are special subclass of the generalized superclass.

For example consider two entity types CAR and TRUCK. They have several common charasteristics like Vehicle-No, Engine-No, Price etc. They can be generalized into an entity type VEHICLE.

Now, the CAR and TRUCK entity types are subclasses of the generalized superclass VEHICLE.

So, Generalization is a process of defining a generalized entity type from the given entity types.

Constraints on specialization

Basically there are three constraints on specialization.

(I) Disjointness Constraints

(II) Overlapping Constraint

(III) Completeness Constraint (which can be devided into total specialization constraint and partial specialization constraint)

As disjoint and completeness constraints are independent, we have four possible constraints on specialization.

(I) Disjoint, total (II) Disjoint, partial

(III) Overlapping, total (IV) Overlapping, partial

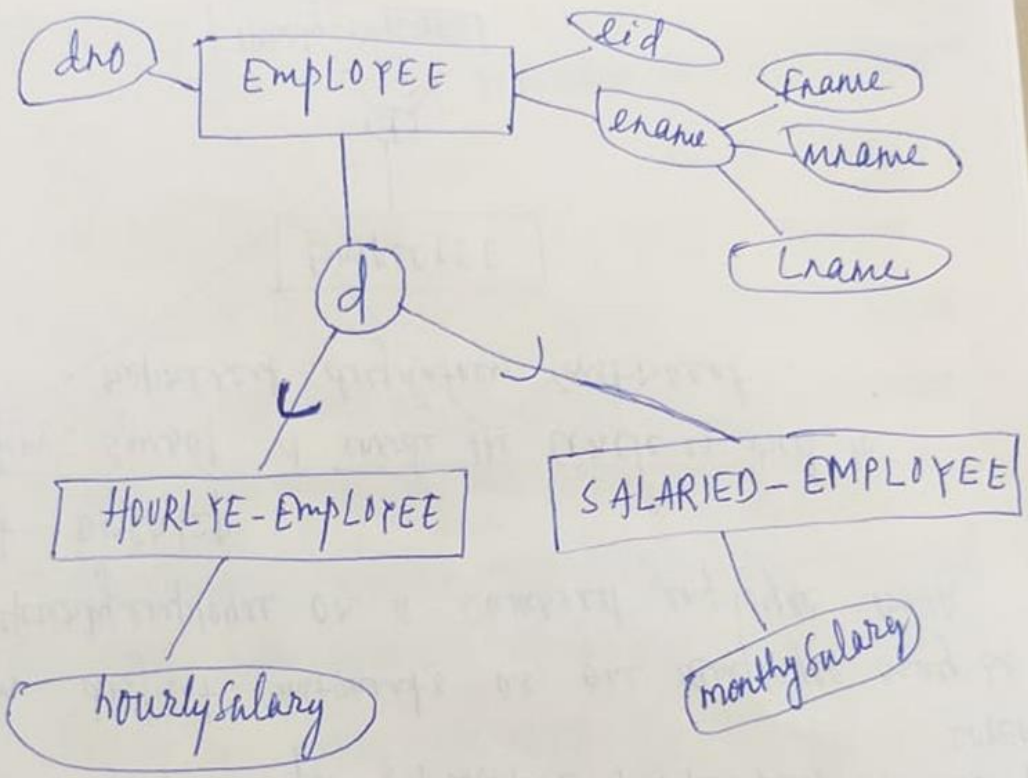
Constraints on specialization or generalization.

Disjoint Constraint or Disjointness Constraint

Disjointness constraint specifies that the subclasses of the specialization must be disjoint. This means that an entity can be a member of atmost one of the subclass of the specialization.

for example {HOURLY_EMPLOYEE, SALARIED_EMPLOYEE} form disjoint constraints as one employee may be a hourly employee or a salaried employee. Both can't overlap.

e.g.



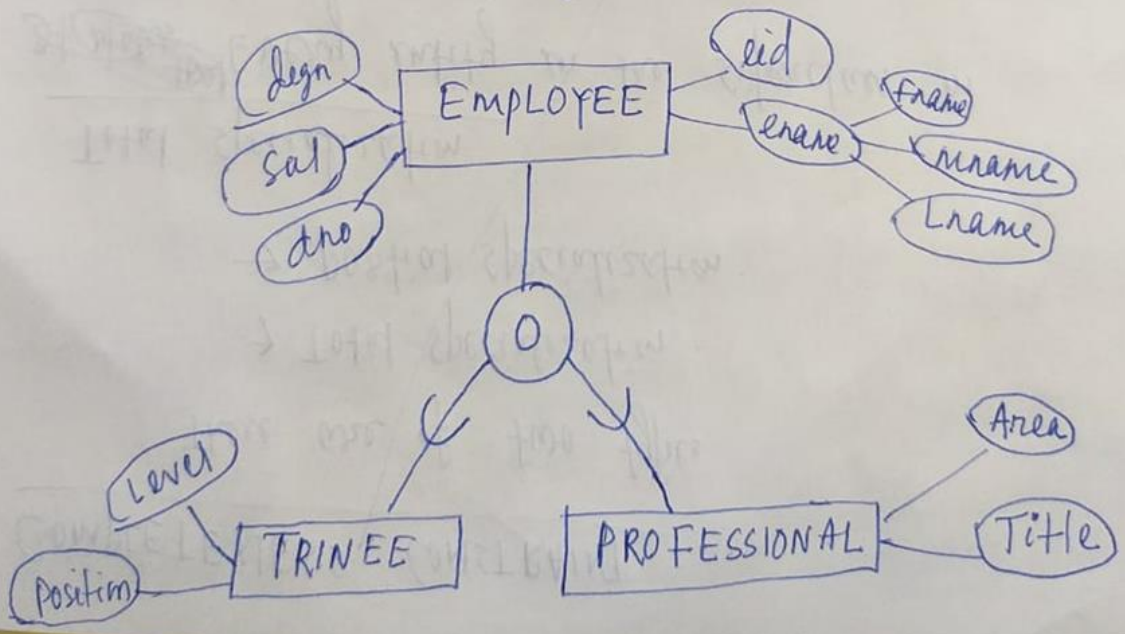
Symbol d inside the circle (d) represent disjointness constraints.

Overlapping Constraint:

Overlapping Constraint specifies that the same entity may be a member of more than one subclass of the specialization. The entities are not disjoint.

e.g. An EMPLOYEE may be a trainee or a professional. Some professionals are also trainees.

Notation: Symbol O inside the circle (O) represents ~~this~~ overlapping constraints.



COMPLETENESS CONSTRAINT

These are of two types

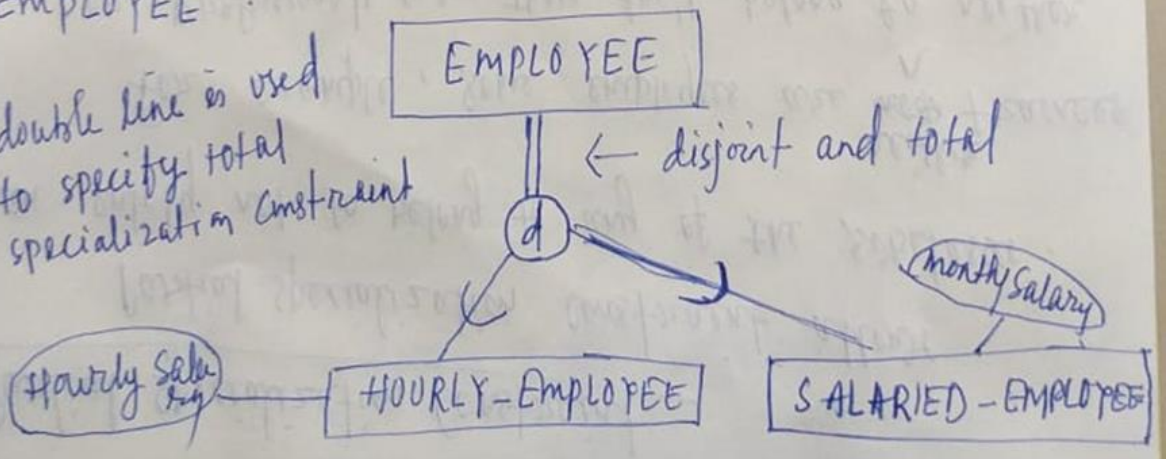
(i) Total specialization constraints.

(ii) Partial specialization Constraint

Total specialization Constraint

It specifies that every entity in the superclass must be a member of atleast one subclass in the specialization.

e.g. An employee must be either Salaried employee or Hourly paid employee. So SALARIED-EMPLOYEE and HOURLY-EMPLOYEE have total specialization constraint with respect to the specialization EMPLOYEE.



double line is used to specify total specialization constraint

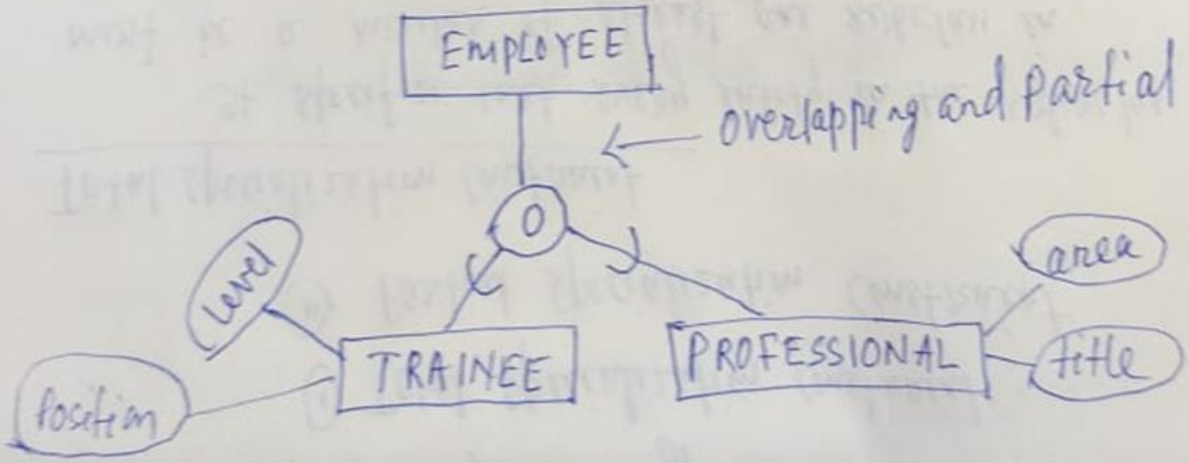
← disjoint and total

Partial specialization Constraint

Partial specialization constraint allows an entity not to belong to any of the subclasses.

for example, some employees are ~~not~~ neither trainees nor professionals i.e. they don't belong to neither TRAINEE entity type nor PROFESSIONAL entity types so, they have partial specialization with respect to EMPLOYEE entity type i.e. the superclass.

of is represented by single line.



INHERITANCE

Inheritance is a process where an entity of a subclass inherits all the attributes of the entity of the superclass. When a subclass inherits a superclass, the entites of subclass have all the attributes of entity of superclass as a member plus values for its specific attribute.

The entity in a subclass also inherits all the relationship in which a superclass participates.

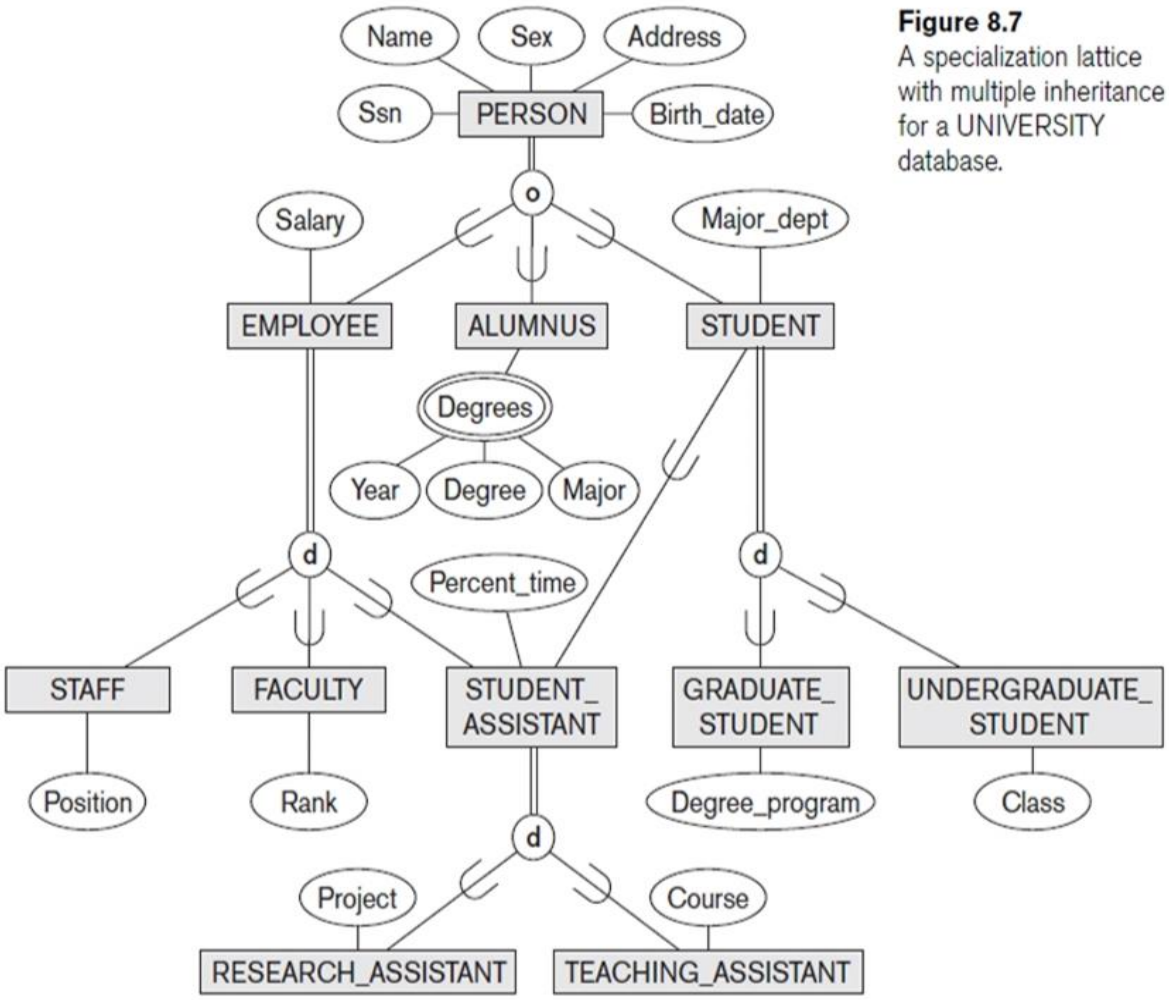


Figure 8.7
A specialization lattice with multiple inheritance for a UNIVERSITY database.