

Unit II: Data Modeling and Database Design

Chapter 04: Enhanced Entity Relationship (EER) Model Concepts and Constraints

Enhanced ER(EER) Model Concepts

- # The enhanced data modeling concepts such as subclasses/superclasses, specialization/generalization, categories, attribute inheritance are incorporated into a basic ER-Model concepts.
- # The resulting model is called the enhanced-ER or Extended ER (E2R or EER) model
- # It is used to model the complex database applications more completely and accurately
- # Example of complex database applications:
CAD/CAM, GIS, complex software system, etc.....

1. Subclasses and Superclasses

- # An entity type may have additional meaningful subgroupings of its entities
E.g: EMPLOYEE may be further grouped into:
SECRETARY, ENGINEER, TECHNICIAN,...
 - Based on the Employee's job

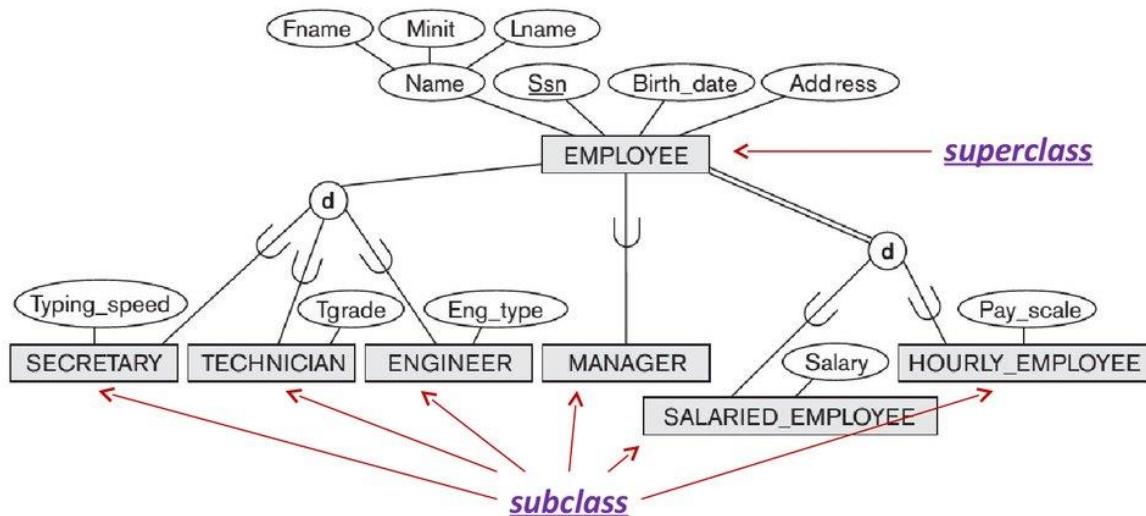
MANAGER

- Employee who are manager

SALARIED_EMPLOYEE, HOURLY_EMPLOYEE

- Based on the Employee method of pay
- ✓ Each of these groupings is a subset of EMPLOYEE entities
 - ✓ Each is called a subclass of EMPLOYEE
 - ✓ EMPLOYEE is the superclass for each of these subclasses
- + Their relationship are called super-class/sub-class relationships.
 - + The subclass/superclass relationships are also called **IS-A(or IS-AN)** relationships
E.g: SECRETARY **IS-A** EMPLOYEE,
TECHNICIAN **IS-A** EMPLOYEE, ...).

Subclasses and Superclasses



2. Attribute inheritance in Superclass/Subclass Relationship

- ✚ An entity which is a member of a subclass represents the same real-world entity as some member of the superclass
- ✚ An entity of a subclass *inherits* all attributes of the superclass entity
- ✚ It also inherits all relationships

3. Specialization

- ✚ Is the process of defining a set of subclasses of a superclass

- ✚ The set of subclasses is defined based upon some distinguishing characteristics of the entities in the superclass

E.g: { SECRETARY, ENGINEER, TECHNICIAN } is a specialization of EMPLOYEE based upon *job type*.

- May have several specializations of the same superclass

E.g: Another specialization of EMPLOYEE based on *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 attributes.
e.g: TypingSpeed of SECRETARY
- The subclass can participate in specific relationship types.

E.g: BELONGS_TO of HOURLY_EMPLOYEE

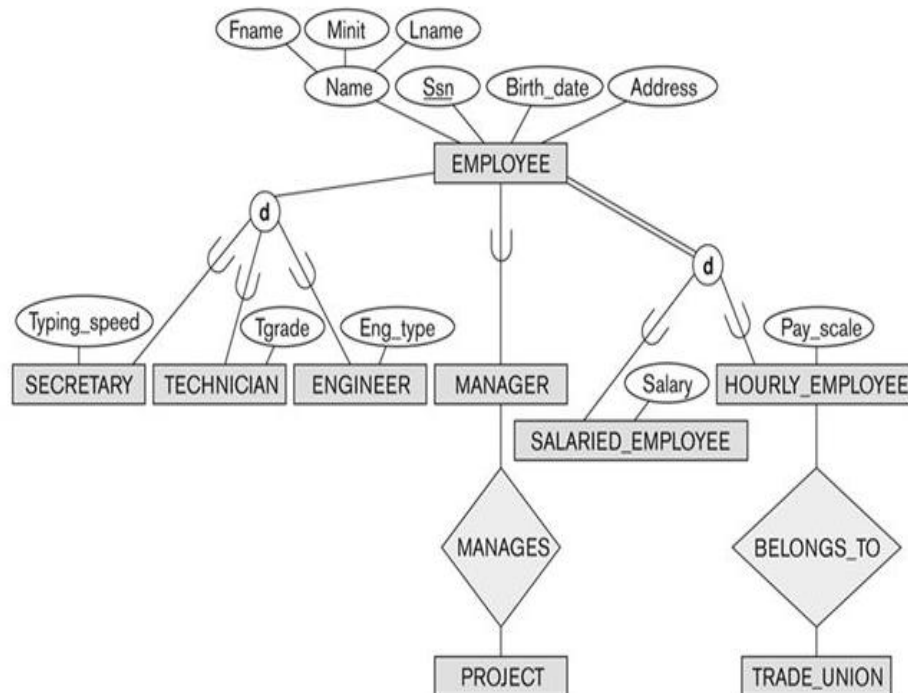


Figure: Example of Specialization

4. Generalization

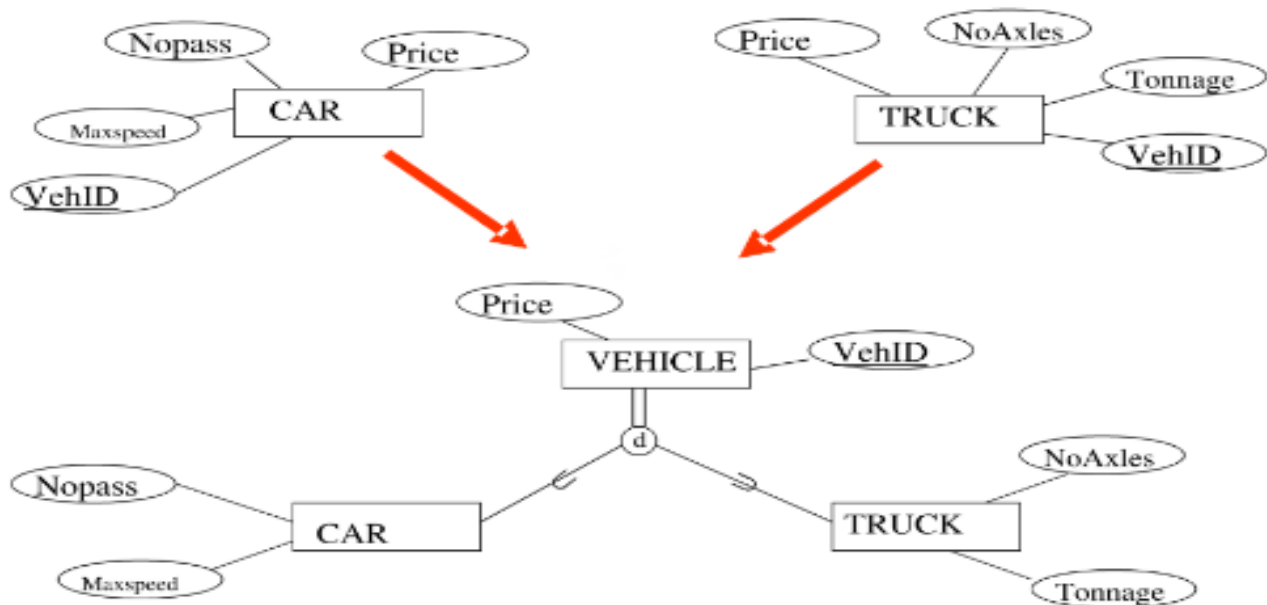
- ✚ Refer the process of defining a generalized entity type from a set of entity types i.e the reverse of the specialization process
- ✚ Several classes with common features are generalized into a superclass; original classes become its subclasses

E.g: CAR, TRUCK are 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

E.g: {SECRETARY, ENGINEER, TECHNICIAN} are generalized into EMPLOYEE

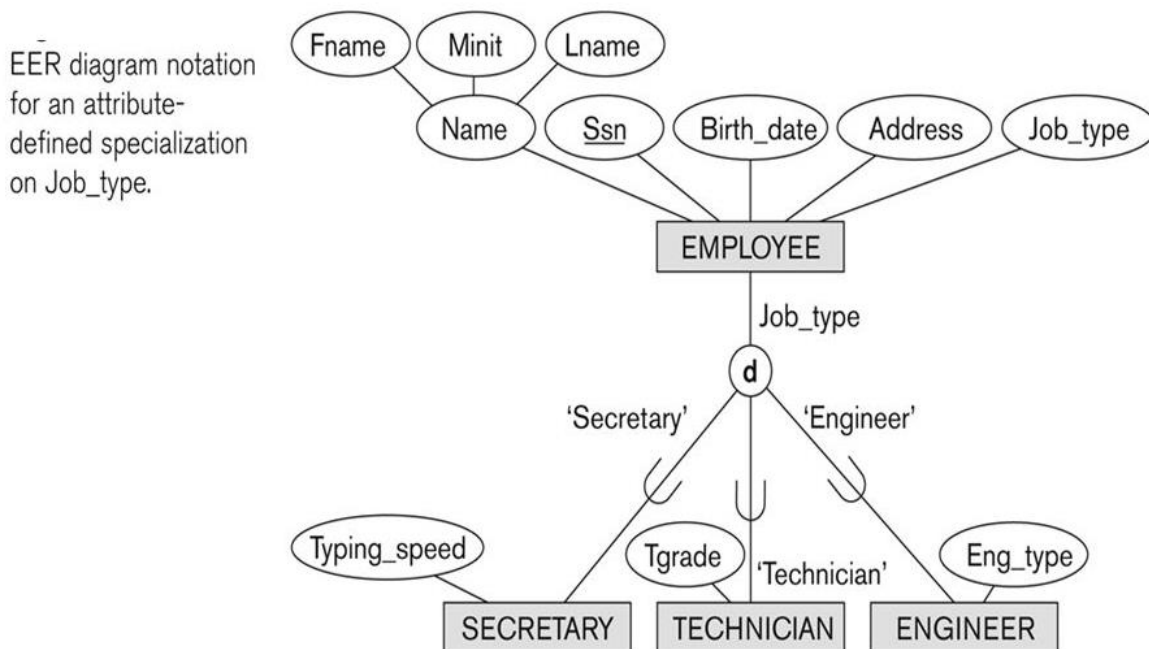


Constraints on Specialization and Generalization

1. Membership Constraint: specifies whether the specialization /generalization has membership condition or not

- **Attribute defined-specialization:** specifies that all subclasses must have membership condition on same attribute of the superclass in the specialization
 - ✓ Attribute is called the defining attribute of the specialization

- ✓ E.g: JobType is the defining attribute of the specialization{SECRETARY, TECHNICIAN, ENGINEER} of EMPLOYEE
- If no condition determines membership of the subclass in a specialization, it is called ***user-defined specialization***
 - ✓ 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



2. Disjointness Constraint: specifies whether the specialization/generalization is **disjoint** or **overlap**.

✚ Disjoint Specialization / Generalization:

- Specifies that the subclasses of the specialization must be disjoint (i.e an entity can be member of at most one of the subclasses of the specialization)
- Specified by **d** in EER diagram

Overlap(Non-Disjoint) Specialization/Generalization:

- Specifies that the subclasses of the specialization must be overlapped (i.e the same entity may be a member of **more than one** subclass of the specialization)
- Specified by **o** in EER diagram

Example of Disjoint Partial Specialization

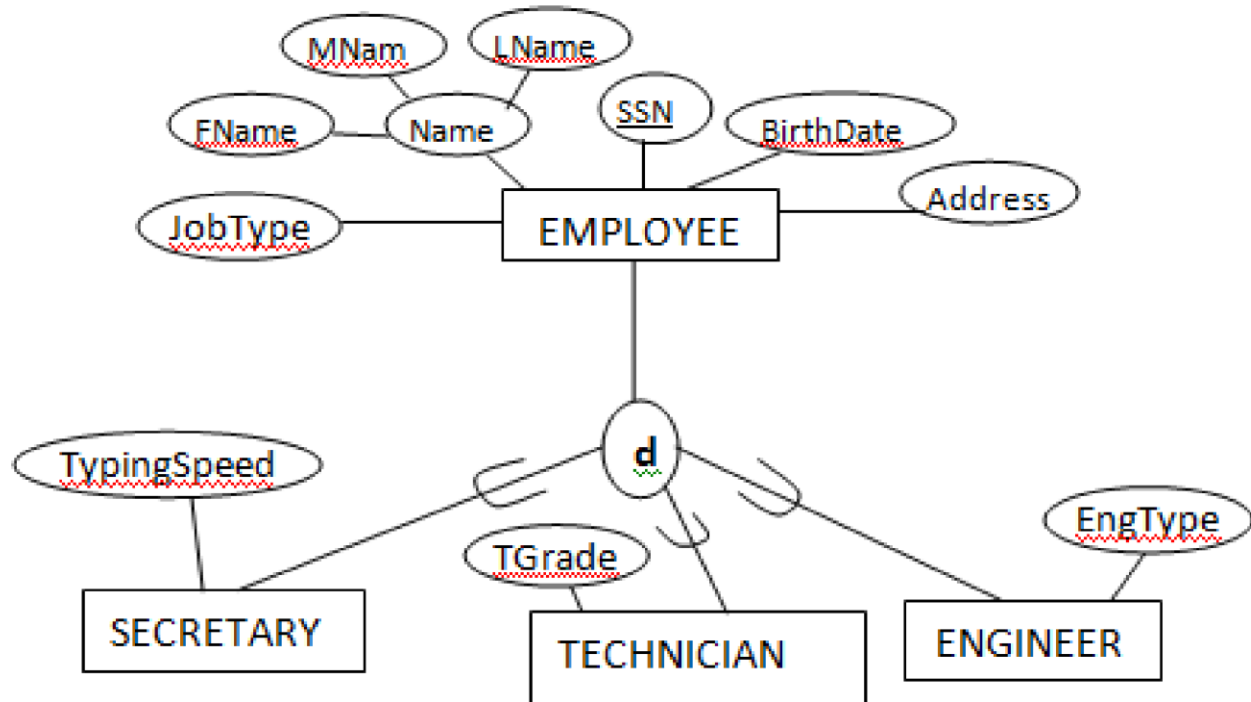


Fig :EER Diagram for a Disjoint-Partial Specialization

Example of Overlapping Specialization

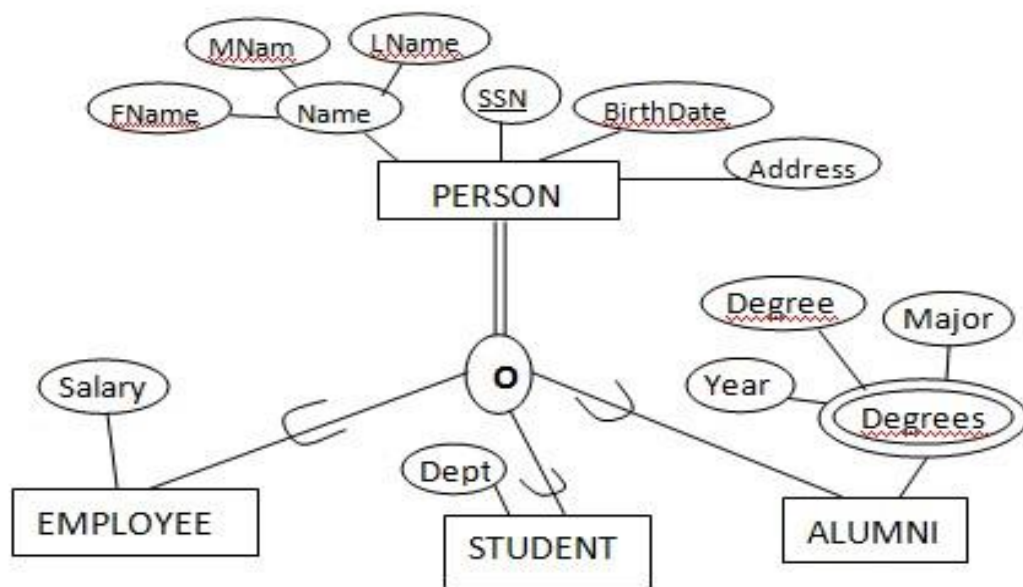


Fig : EER Diagram for an Overlapping(Non-disjoint)-Total Specialization

3. Completeness Constraint: specifies whether the specialization/generalization is **Total** or **Partial**.

- ✚ **Total Specialization/Generalization:**
 - ✓ specifies that **every entity** in the superclass **must be a member of at least one of the subclasses**
 - ✓ Specified by a **double line** in EER diagrams
- ✚ **Partial Specialization/Generalization:**
 - ✓ allows **an entity not to belong** to any of the subclasses
 - ✓ Specified by a **single line** in EER diagrams

Hence, we have four possible constraints on specialization/generalization:

- ✓ Disjoint, total
- ✓ Disjoint, partial
- ✓ Overlapping, total
- ✓ Overlapping, partial
- Note: Generalization usually is total because the superclass is derived from the subclasses.

Specialization / Generalization Hierarchies, Lattices and Shared Subclasses

- ✓ A subclass may have its own subclasses and forms a **specialization 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**

Specialization Vs Generalization (key point)

- In specialization, start with an entity type and then define subclasses of the entity type by successive specialization (top down conceptual refinement process)
- In generalization, start with many entity types and generalize those that have common properties (bottom up conceptual synthesis process)

In practice, the combination of two processes is employed

Specialization / Generalization Lattice Example-1

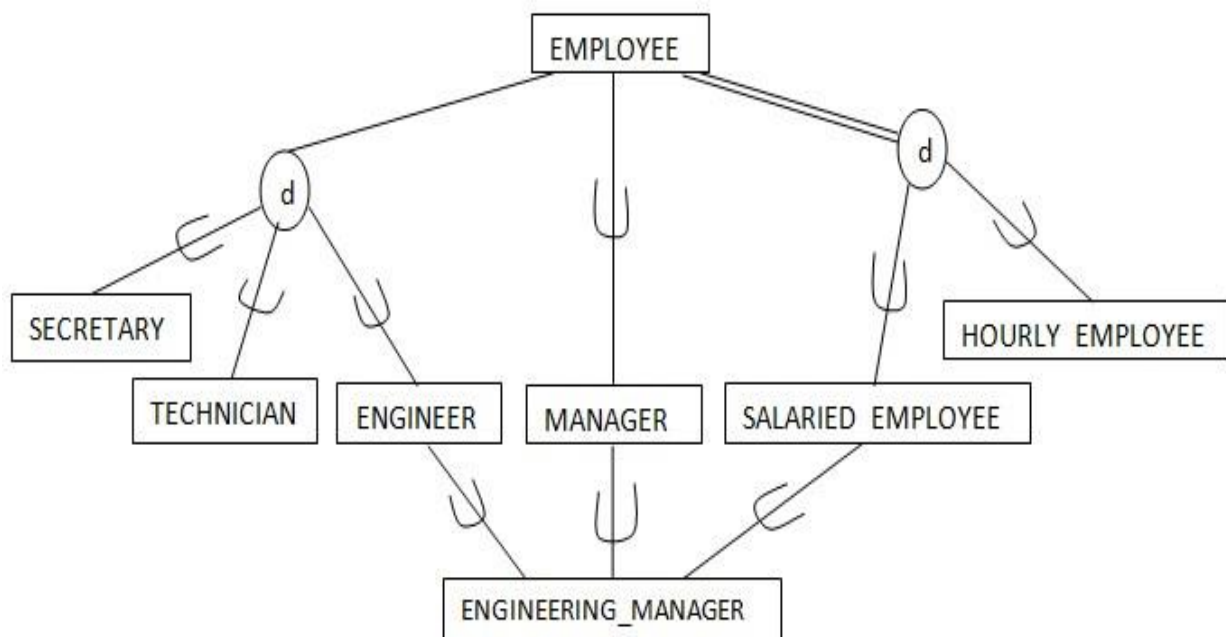


Fig:-A Specialization hierarchy & Lattice with shared subclass ENGINEERING MANAGER

Specialization / Generalization Lattice Example- (UNIVERSITY)

