Chapter 4: Enhanced Entity-Relationship (EER) Modeling

Outline:

4.1 Subclasses and Superclasses

4.2 Specialization

4.3 Generalization

4.4 Constraints on Specialization and Generalization
- Membership Constraints.
- Belonging Constraints.
- Completeness Constraints.

4.5 An Example of Patient EER Schema

4.1 Subclasses and Superclasses

- An entity type may have additional meaningful subgroupings of its entities
- Example: EMPLOYEE may be further grouped into:
  - SECRETARY, ENGINEER, TECHNICIAN, Based on the EMPLOYEE's Job
  - MANAGER
  - EMPLOYEEs who are managers SALARIED_EMPLOYEE, HOURLY_EMPLOYEE Based on the EMPLOYEE's method of pay

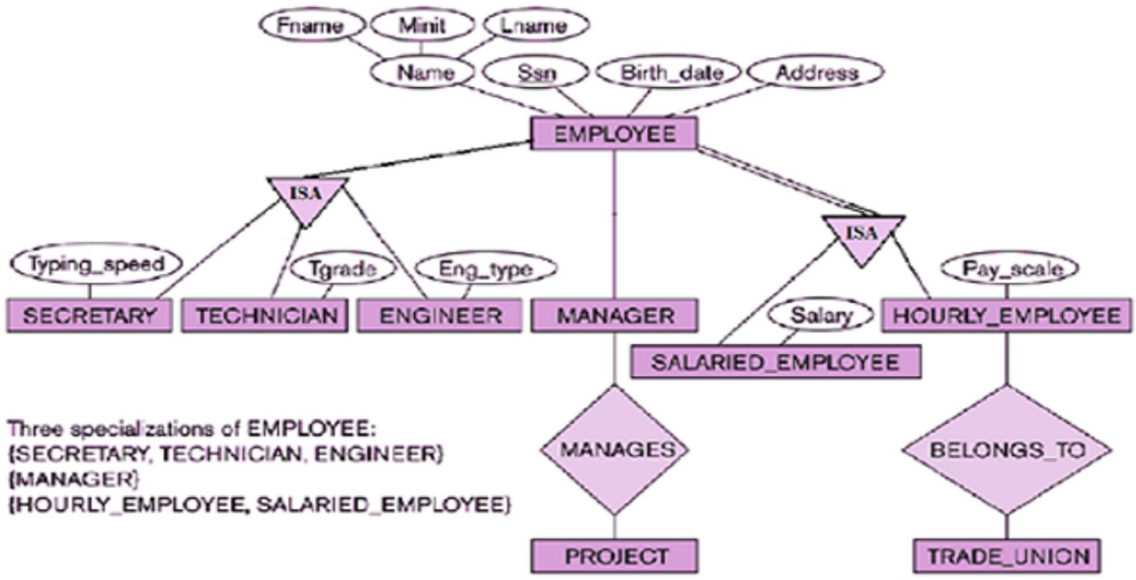
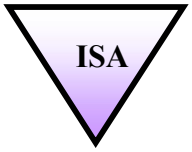


Figure 4.1: EER Model

- EER diagrams extend ER diagrams to represent these additional subgroupings, called subclasses or subtypes

  - Each of these subgroupings is a subset of EMPLOYEE entities
  - Each is called a **subclass (low level entity)** of EMPLOYEE
  - EMPLOYEE is the **superclass (high level entity**) for each of these subclasses
  - These are called **superclass/subclass** relationships:
    - EMPLOYEE/SECRETARY
    - EMPLOYEE/TECHNICIAN
    - EMPLOYEE/MANAGER
  - These are also called **IS-A relationships**
    SECRETARY **IS-A** EMPLOYEE, TECHNICIAN **IS-A** EMPLOYEE, ….

  Pictured by triangle:



- **Note:** An entity that is member of a subclass represents the same real-world entity as some member of the superclass:
  - 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 to any number of its subclasses
  - **For Example: (Figure 4.1)**
    - A salaried employee who is also an engineer belongs to the two subclasses:
      - ENGINEER, and
      - SALARIED_EMPLOYEE
    - A salaried employee who is also an engineering manager belongs to the three subclasses:
      - MANAGER,
      - ENGINEER, and
      - SALARIED_EMPLOYEE
- It is not necessary that every entity in a superclass be a member of some subclass
- An entity that is member of a subclass inherits
  - All attributes of the entity as a member of the superclass
  - All relationships of the entity as a member of the superclass
  - Example:
    - In the previous EER diagram, SECRETARY (as well as TECHNICIAN and ENGINEER) inherit the attributes Name, SSN, …, from EMPLOYEE
    - Every SECRETARY entity will have values for the inherited attributes
- A subclass may itself have further subclasses specified on it. Hierarchy has a constraint that every subclass has only one superclass (called single inheritance); this is basically a tree structure
    - For example superclass Person with two subclasses (student and employee) and the subclasses student is the superclass of graduate and undergraduate student subclasses), the employee is the superclass of three subclasses (secretary, engineer, and technical). As you see in the following EER diagram:

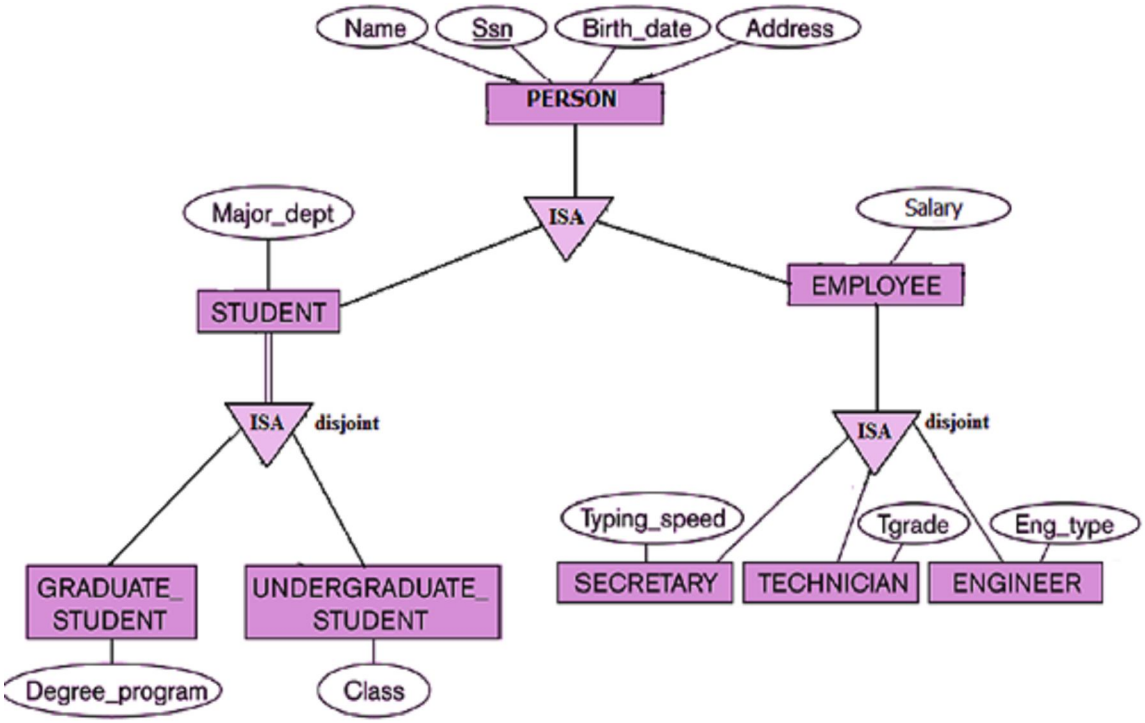


Figure 4.2: Hierarchy EER Model

4.2 Specialization

- Specialization is the process of defining a set of subclasses of a superclass
- In specialization, start with an entity type and then define subclasses of the entity type by successive specialization called a **top down** conceptual refinement process
- 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

4.3 Generalization

- Generalization is the reverse of the specialization process

- In generalization, start with many entity types and generalize those that have common properties called a **bottom up** conceptual synthesis 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
- Show the following two entity type car and truck and generalizing them to vehicle superclass:

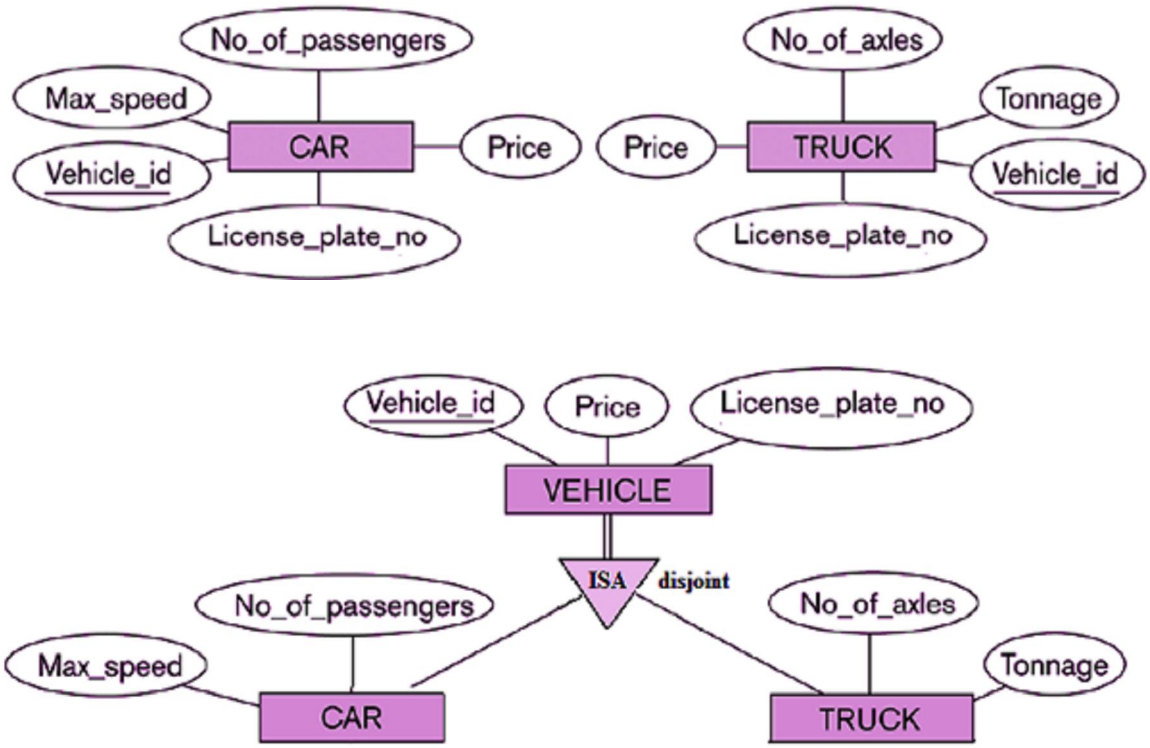


Figure 4.3: Generalization Example

4.4 Constraints on Specialization and Generalization

1. Membership Constraints

- This constraints determining which entities can be members of a given lower-level entity set. Such membership may be one of the following

A. **Condition-defined**: In condition-defined lower-level entity sets, membership is evaluated on the basis of whether or not an entity satisfies an *explicit condition or predicate*. 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

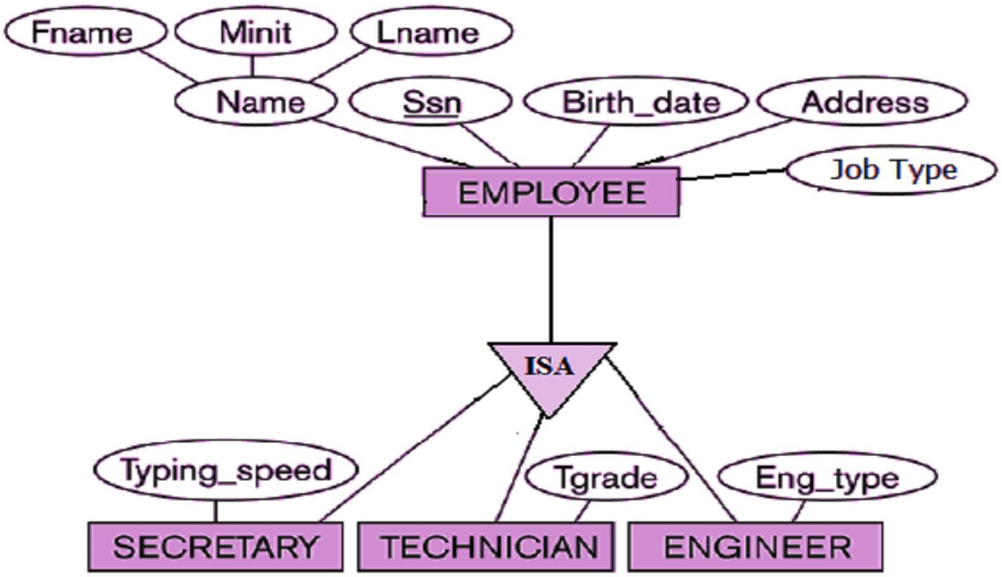


Figure 4.4: Condition-defined membership constraint example

B. **User-defined**: User-defined lower-level entity sets are not constrained by a membership condition; rather, the database user assigns entities to a given entity set. For example, let us assume that, after 3 months of employment, bank employees are assigned to one of four work teams.

2. *Belonging Constraints*

- This constraint relates to whether or not entities may belong to more than one lower-level entity set within a single generalization. The lower-level entity sets may be one of the following:

:

A. Disjoint Constraint:

- A disjointness constraint requires that an entity belong to no more than one (at most one) lower-level entity set (subclasses).

- In an EER diagram by adding the word ***disjoint*** next to the triangle symbol.
- For example, a vehicle entity; an entity can be either a car or a truck, but cannot be both

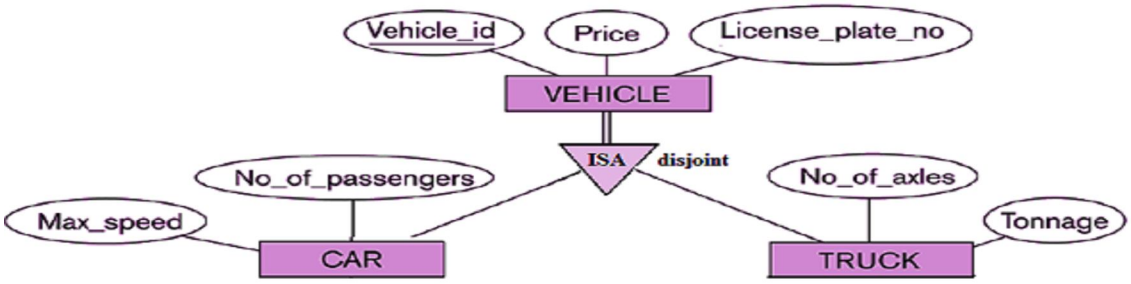


Figure 4.5: Disjoint constraint example

B. Overlapping Constraint:

- That is the same entity may be a member of more than one subclass.
- Lower-level entity overlap is the default case
- A given person may therefore appear in more than one of the lower-level entity sets of person; an employee can also be a student.

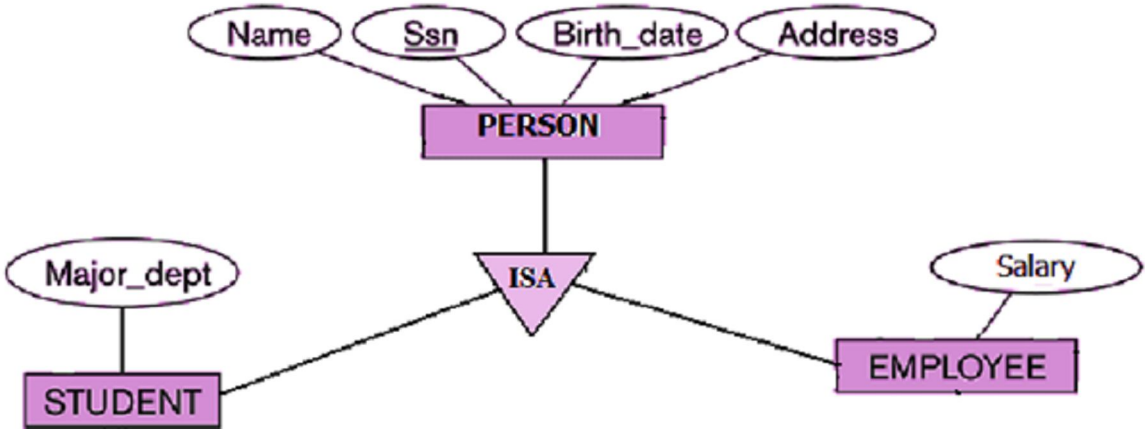


Figure 4.6: Overlapping constraint example

3. Completeness Constraints

- Specifies whether or not an entity in the higher-level entity set must belong to at least one of the lower-level entity sets

A. Total

- Specifies that every entity in the superclass must be a member of some subclass.
- Shown in EER diagrams by a double line

B. Partial

- Allows an entity not to belong to any of the subclasses
- Partial generalization is the default ;shown in EER diagrams by a single line

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

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

4.5 An Example of Patient EER Schema

Consider the following information about a patient database:

- A patient is identified by patient id, and admission date.
- A patient must be either an Emergency patient or a resident patient.
- We must record the checkback date for Emergency patient.
- Only resident patient is assigned to one room. Room has a unique number, degree, and floor number.
- For each room there are a number of beds each of which has a bed number, specification. If a room is deleted, you need not keep track of its beds any longer.
- Both emergency patient and resident are cared for by a responsible Doctor. Doctor is identified by Doctor number. For each Doctor, the name, several specialty, and years of experience must be recorded.
- Doctors prescribe medicine for patients. Medicine is identified by trade name that has scientific name and formula. The medicine depends on another medicine to be taking before.
- A doctor could prescribe one or more drugs for several patients, and a patient could obtain prescriptions from several doctors. Each prescription has a date and a quantity of medicine.

Draw an ER/EER diagram that captures the above information.