Extended ER features

T. JOSHVA DEVADAS

PROFESSOR

SJT ANNEX 103F

- EER is a high-level data model that incorporates the extensions to the original ER model.
- **It is a diagrammatic technique for displaying the following concepts**
  - Sub Class and Super Class
  - Specialization and Generalization
  - Union or Category
  - Aggregation
- These concepts are used when the comes in EER schema and the resulting schema diagrams called as EER Diagrams.

A. Sub Class and Super Class

- Sub class and Super class relationship leads the concept of Inheritance. The relationship between sub class and super class is denoted with symbol a **hollow arrow-head pointing from the specialized entity to the other entity**
- **Super Class :** Super class is an entity type that has a relationship with one or more subtypes.
- An entity cannot exist in database merely by being member of any super class. For example: **Shape** super class is having sub groups as **Square, Circle, Triangle**.
- **Sub Class :** Sub class is a group of entities with unique attributes.
- Sub class inherit properties and attributes from its super class. **For example:** Square, Circle, Triangle are the sub class of Shape super class.

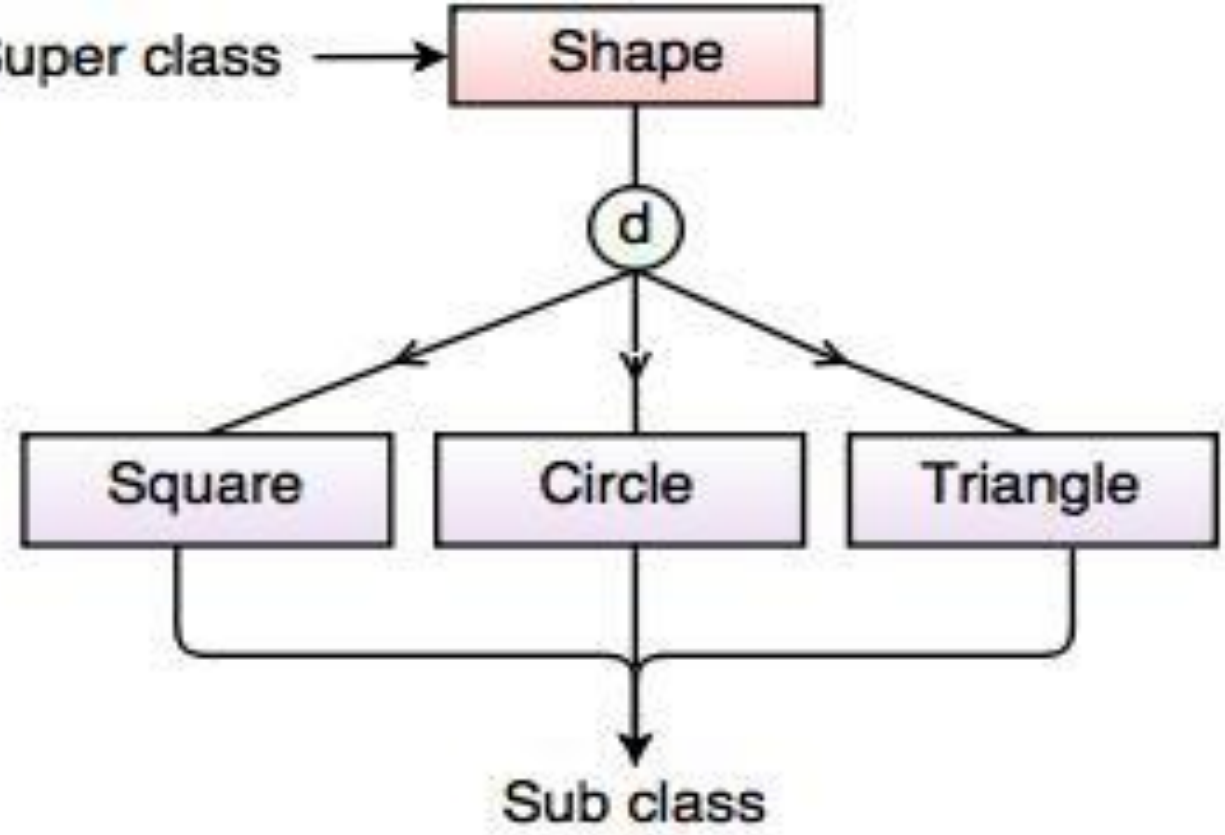
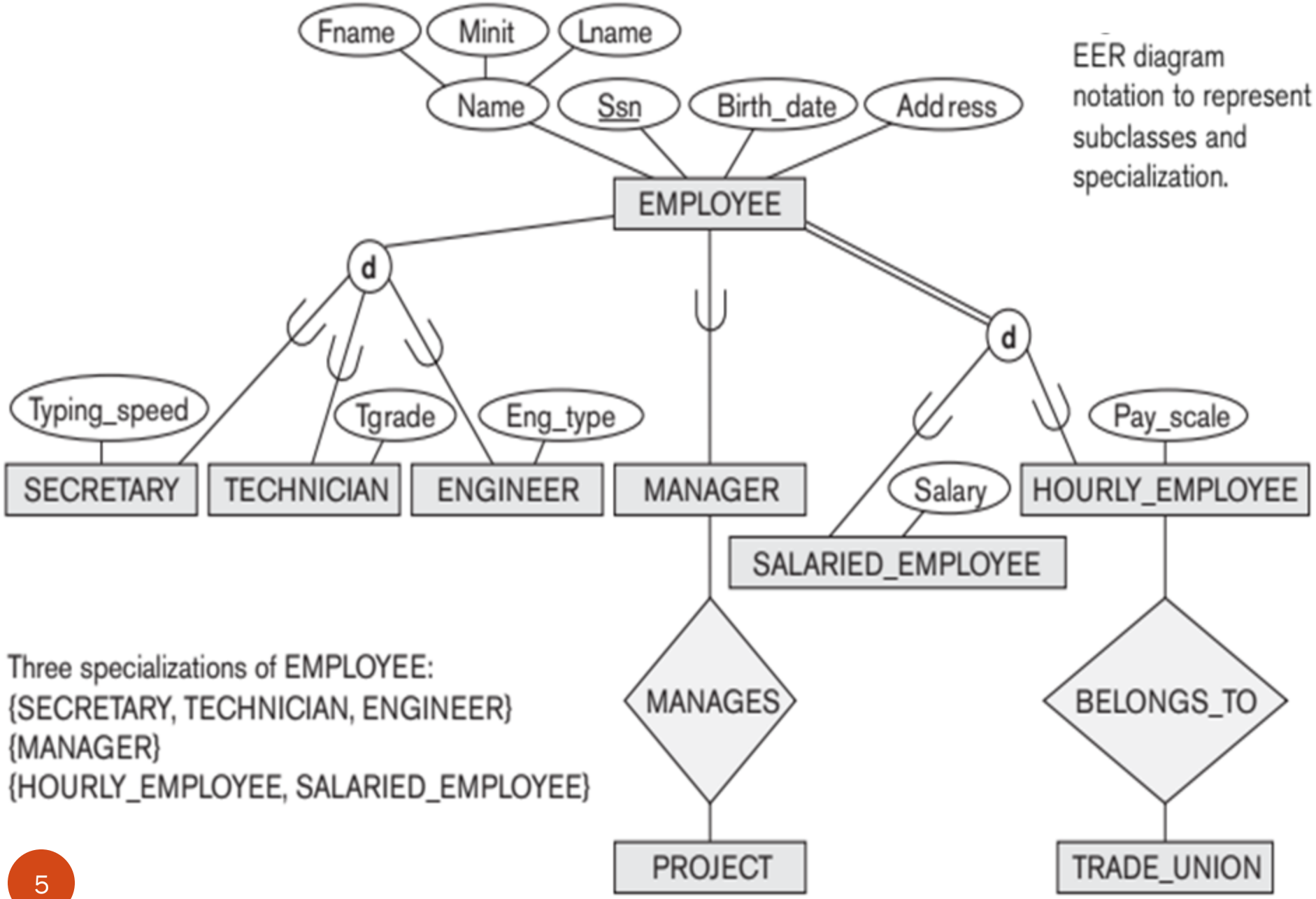


Fig. Super class/Sub class Relationship



EER diagram notation to represent subclasses and specialization.

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

Constraints and its Types

- There are three constraints that may apply to a specialization/generalization:
  - **Disjoint Constraints and**
  - **Completeness Constraints.**
  - **Membership Constraints**

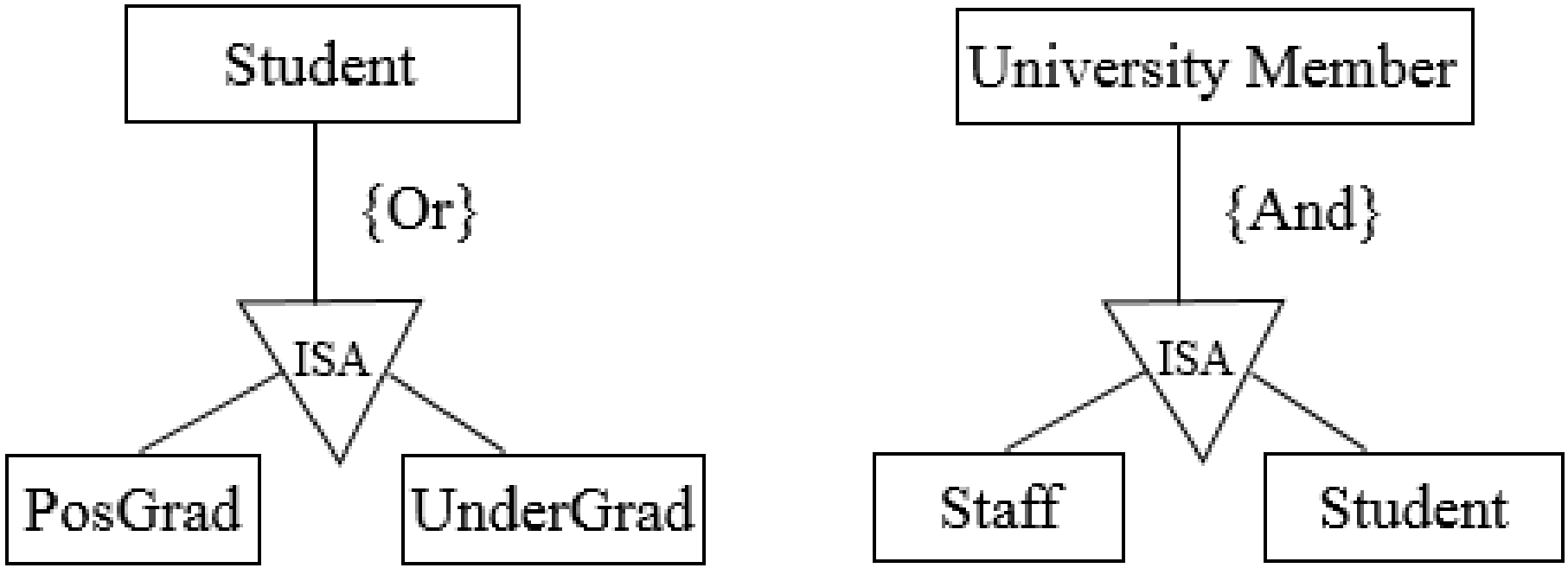
Disjointness Constraints on Specialization and Generalization

- There are two type of constraints in specializations.
- **Disjointness** specifies that the subclasses of the specialization must be disjoint. The notation of disjoint constraint is

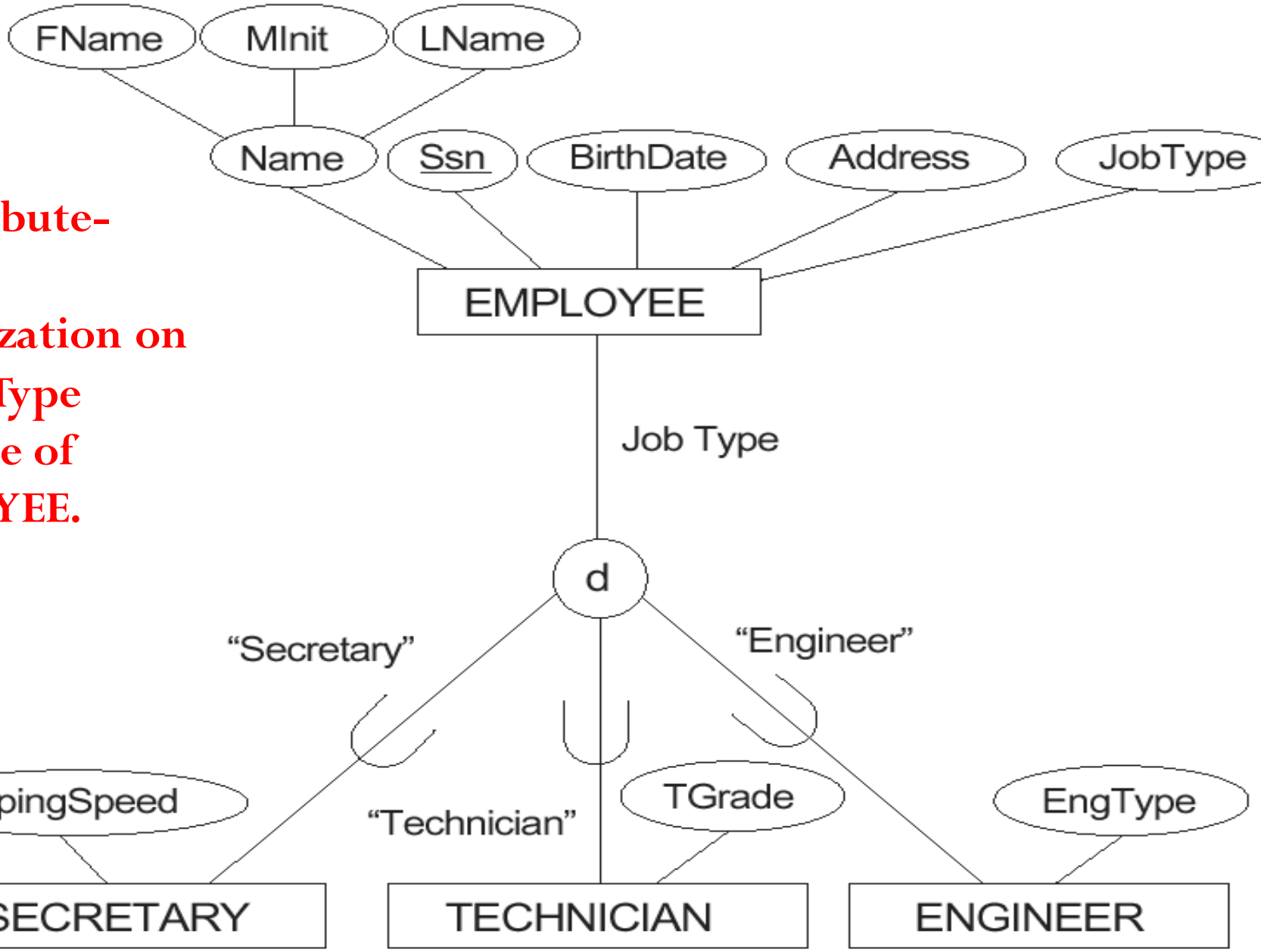
- There are four further possible constraints on specialization
  - Disjoint total
  - Disjoint partial
  - Overlapping total
  - Overlapping partial
- **Completeness**
  - Decide whether a supertype instance must also be a member of at least one subtype.

Disjoint Constraint

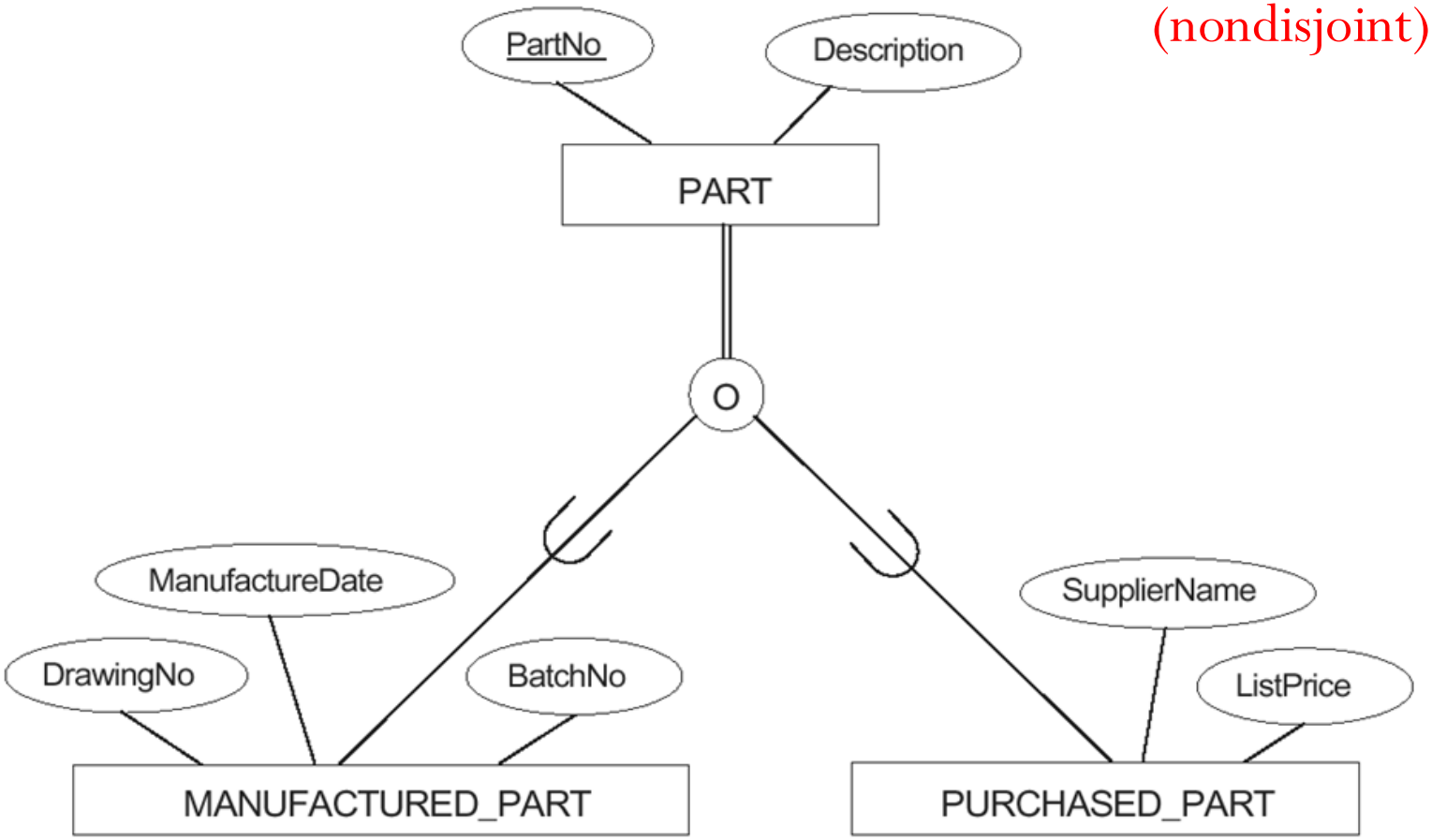
- **Disjoint (d)**
  - The subclasses must have disjoint sets of entities
- **Overlap (o)**
  - The subclasses may have overlapping sets of entities



An attribute-defined specialization on the JobType attribute of EMPLOYEE.

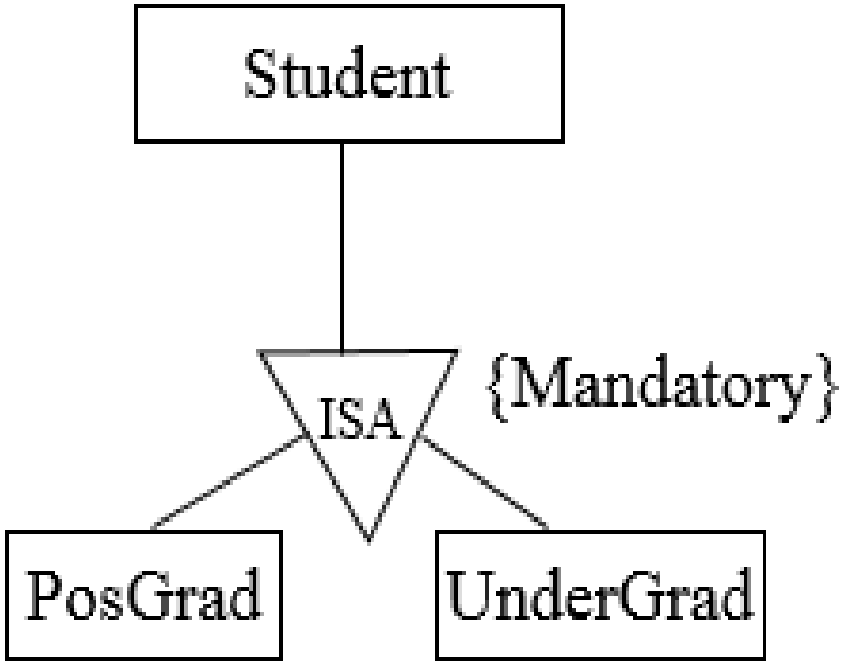


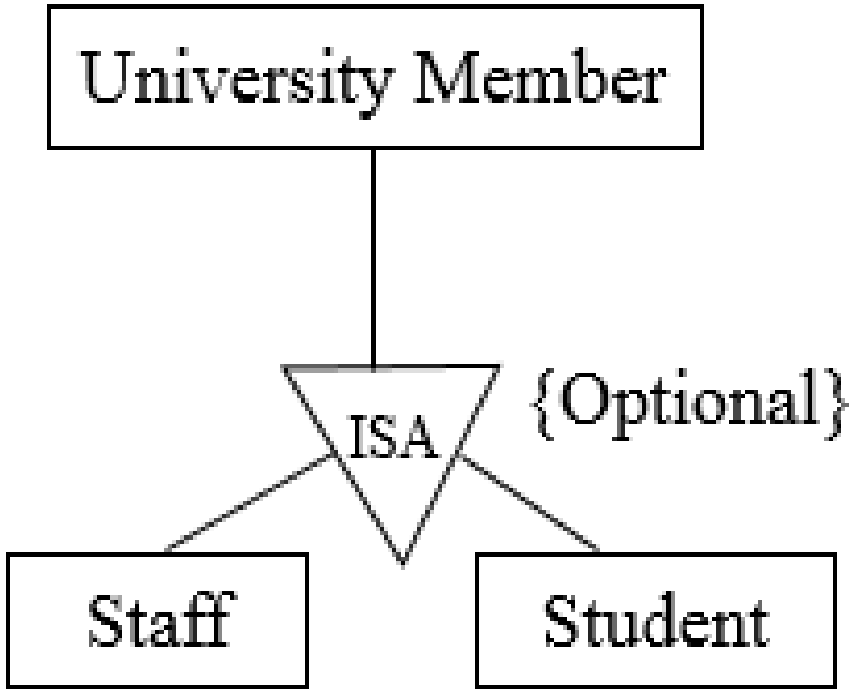
Notation for specialization with overlapping (nondisjoint) subclasses.



Completeness Constraints

- **Partial**
  - An entity may not belong to any of the subclasses (single-line)
- **Total**
  - Every entity in the superclass must be a member of some subclass (double-edge)





Total
(Specialization)

Partial
(Generalization)

Membership Constraints

- **Predicate defined subclasses**
  - The subclass is defined through a predicate on the attributes of the superclass
  - **Attribute defined subclasses**
  - The subclasses in the specialization are all defined by the same attribute of the superclass
  - **User defined subclasses**
  - Membership in the subclasses is determined at the insertion operation level

Specialization

- An entity set may include subgroupings of entities that are distinct in some way from other entities in the set.
- **Specialization** is the process of defining a *set of subclasses* of an entity type. This entity type is called the **superclass** of the specialization.
- It is a **top down approach**, in which one higher entity can be broken down into two lower level entity.
- It maximizes the difference between the members of an entity by identifying the unique characteristic or attributes of each member.

- In terms of an E-R diagram, specialization is depicted by a **hollow arrow-head pointing from the specialized entity to the other entity**

- The specialization of *person* allows us to distinguish among person entities according to whether they correspond to employees or students:

- In general, a person could be an employee, a student, both, or neither.

- We refer to this relationship as the **ISA** relationship, which stands for "**is a**" and represents, for example, that an instructor "**is a**" employee.

- An entities B is said to be a subclass of another entity A. If and only if it shares an "**is-a**" relationship with A.
- Example: *Car "**is-a**" vehicle.*

  *Manager "**is-a**" Employee.*
- An entity of class B is said to be a **specialization** of entities of class A.
- Conversely, entities of class A are **generalization** of the class B entities.

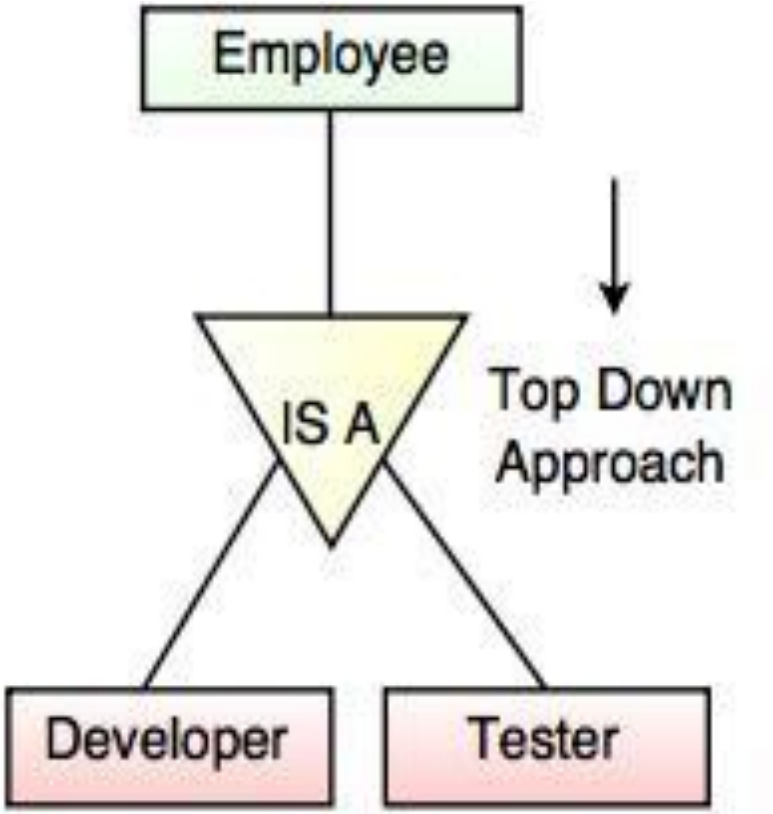
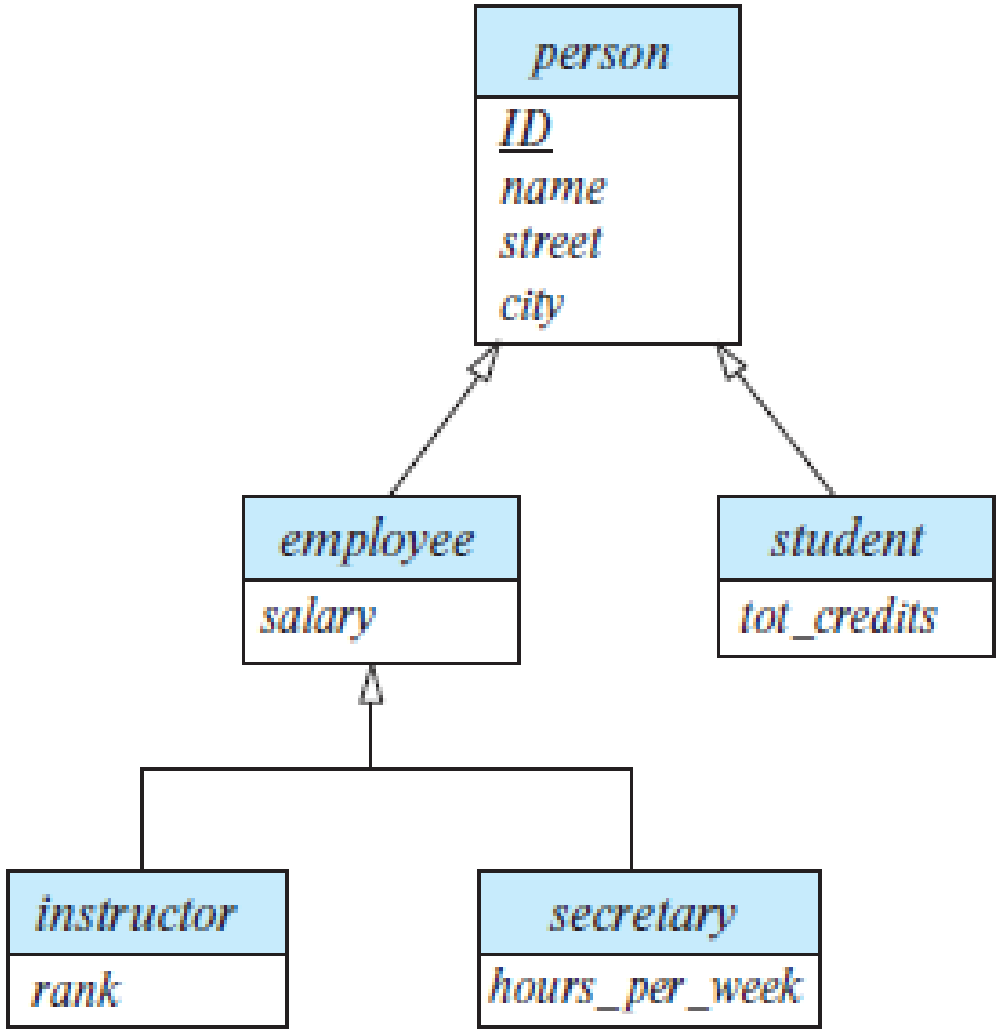
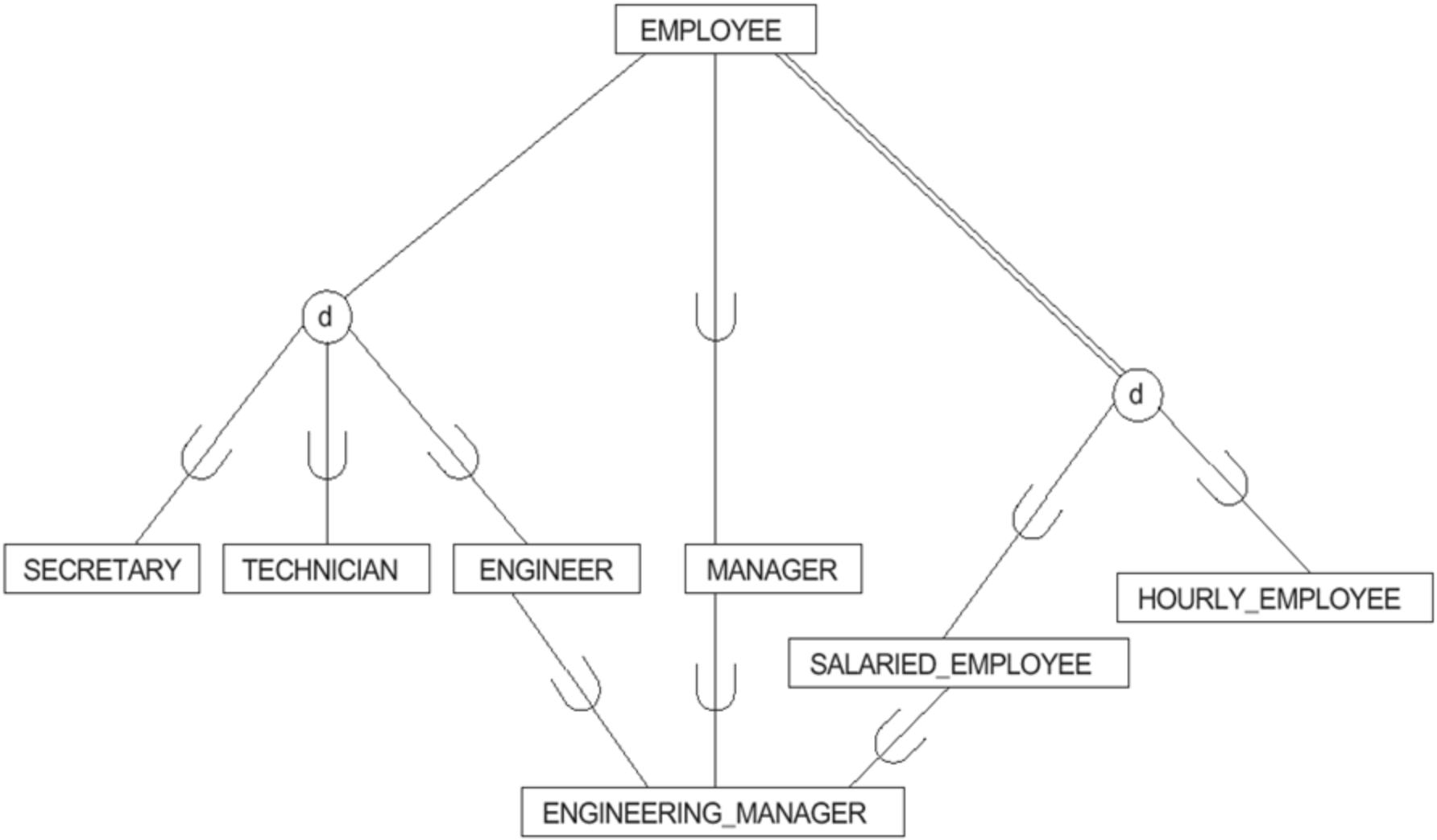


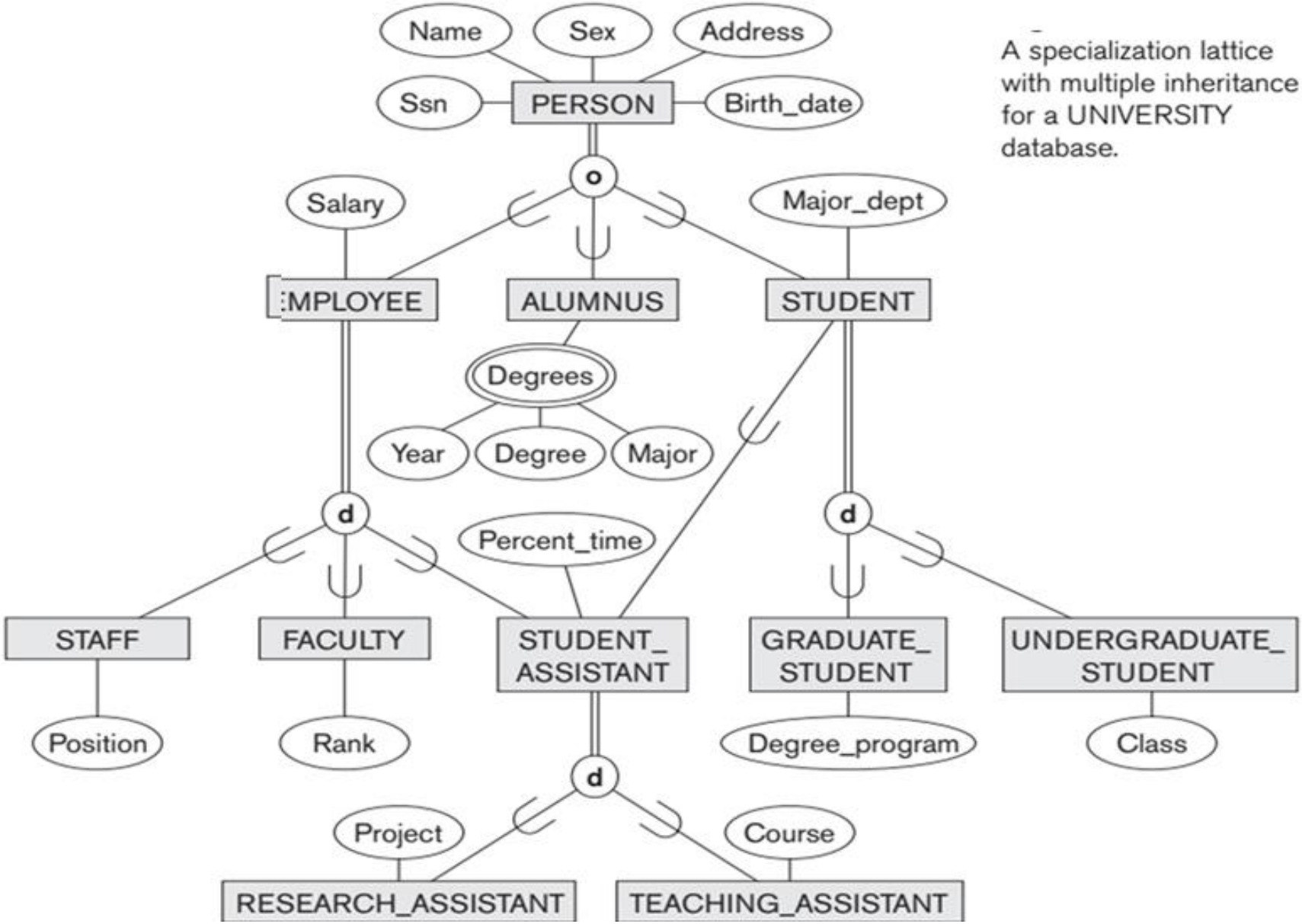
Fig. Specialization



Specialization and generalization.



A specialization lattice with the shared subclass ENGINEERING_MANAGER (with Multiple Inheritance)



A specialization lattice with multiple inheritance for a UNIVERSITY database.

A specialization lattice (with multiple inheritance) for a UNIVERSITY database

Generalization

- Multiple entity sets are synthesized into a higher-level entity set, based on common features (bottom-up).
- Generalization is the process of generalizing the entities which contain the properties of all the generalized entities.
- It is a **bottom approach**, in which two lower level entities combine to form a higher level entity.
- Generalization is the reverse process of Specialization.
- It defines a general entity type from a set of specialized entity type.
- It minimizes the difference between the entities by identifying the common features.

For example:

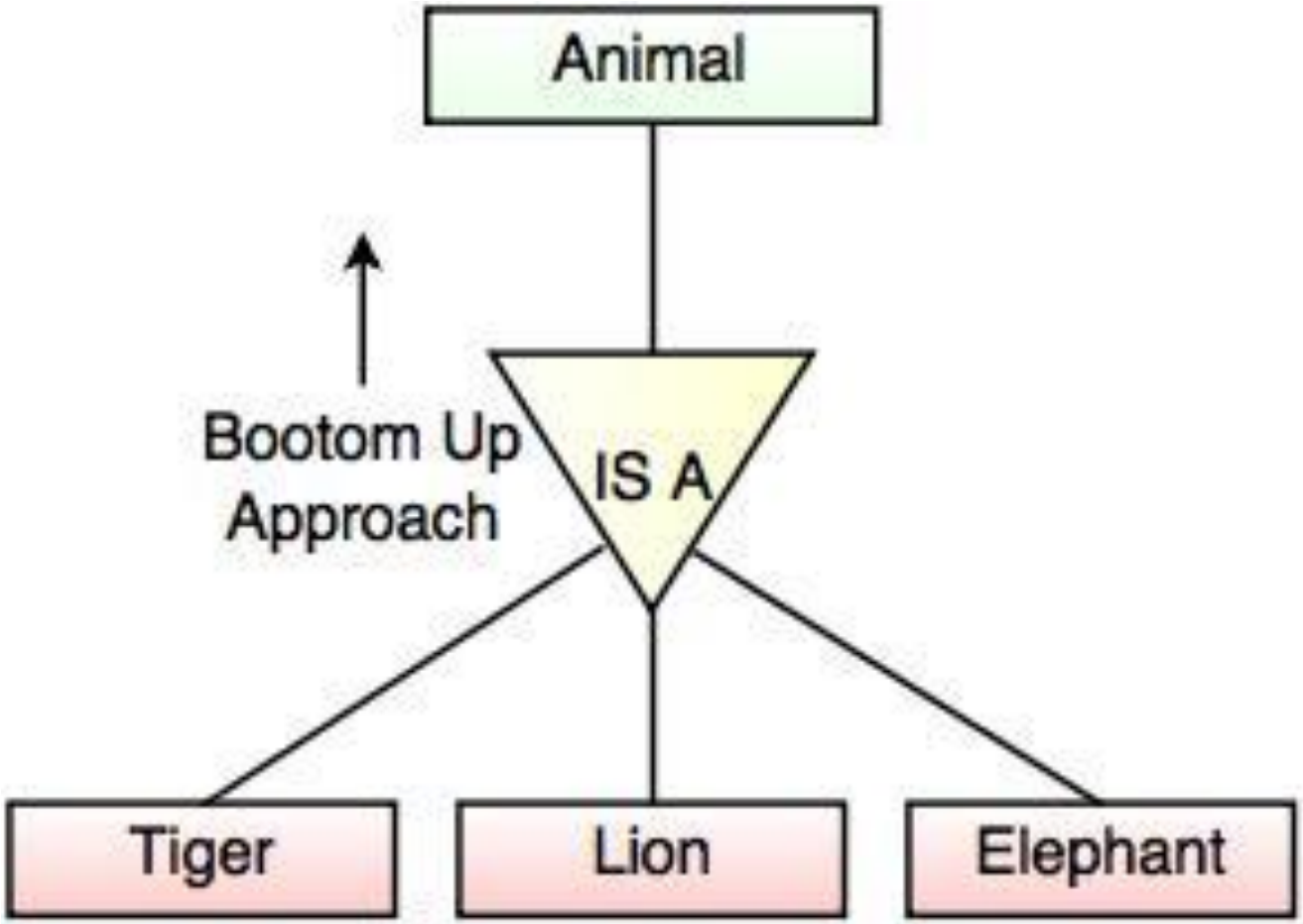


Fig. Generalization

In the example, Tiger, Lion, Elephant can all be generalized as Animals.

Category or Union

- Category represents a single super class or **sub class relationship with more than one super class.**

- It can be a total or partial participation. **For example** Car booking, Car owner can be a person, a bank (holds a possession on a Car) or a company.

- Category (sub class) → Owner is a subset of the union of the three super classes → Company, Bank, and Person. A **Category member must exist in at least one of its super classes.**

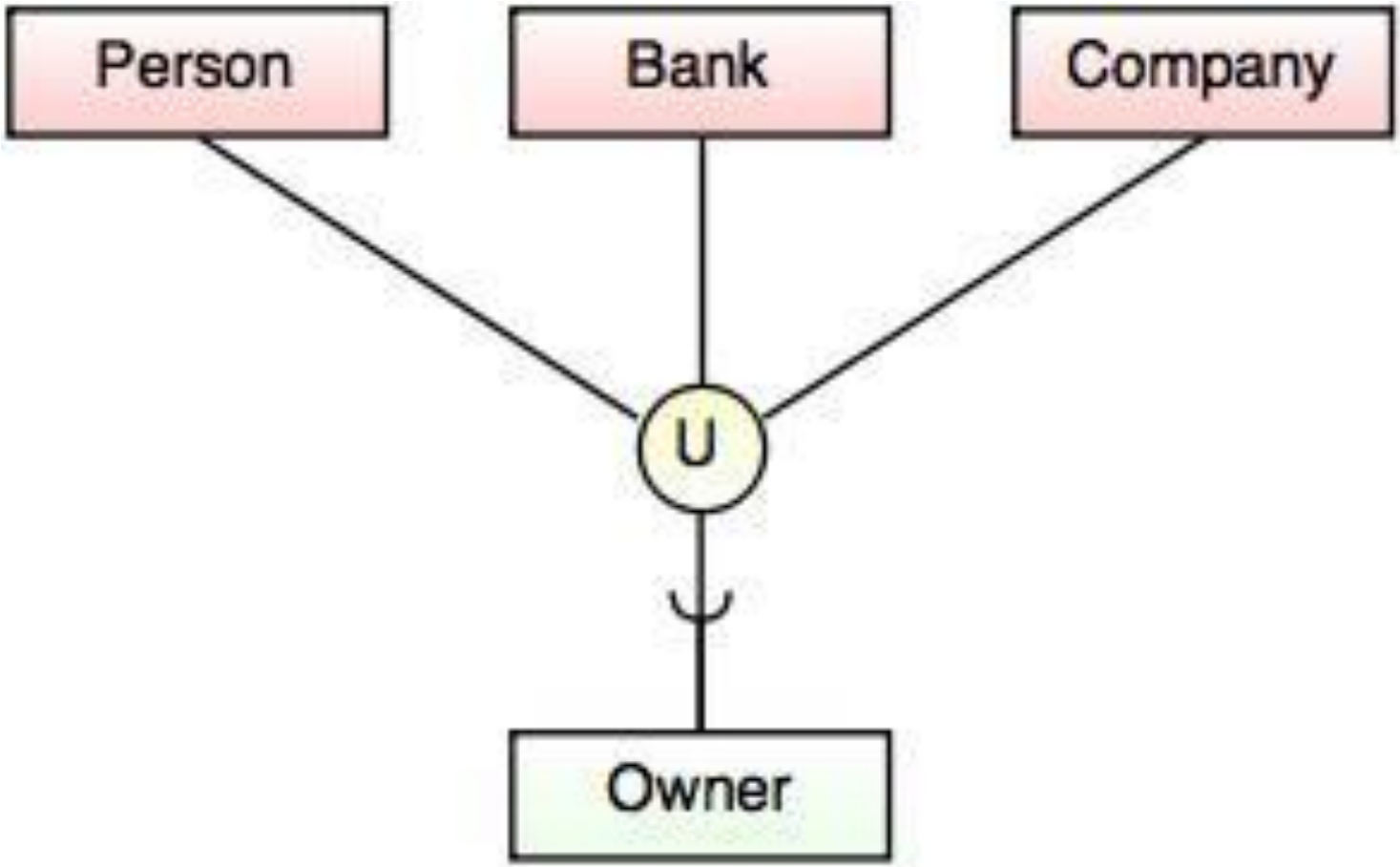


Fig. Categories (Union Type)

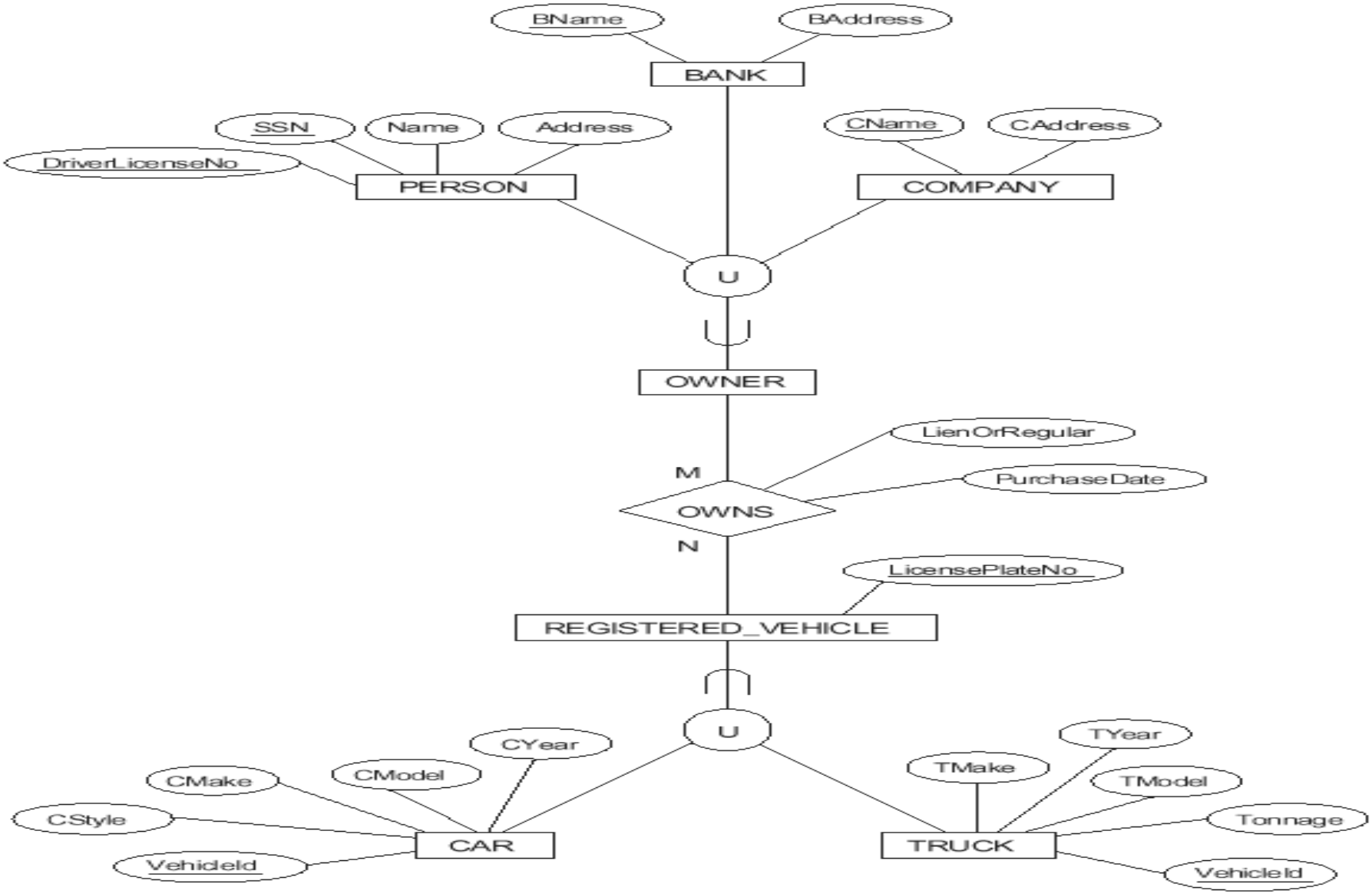


Illustration of how to represent the UNION of two or more entity types/classes using the category notation. Two categories are shown: OWNER and REGISTERED_VEHICLE.

Aggregation

- Aggregation refers to the process by which entities are combined to form a single meaningful entity.
- The specific entities are combined because they do not make sense on their own.
- To establish a single entity, aggregation creates a relationship that combines these entities.
- Aggregation is a process that represent a relationship between a whole object and its component parts.
- It abstracts a relationship between objects and viewing the relationship as an object.
- It is a process when two entity is treated as a single entity.

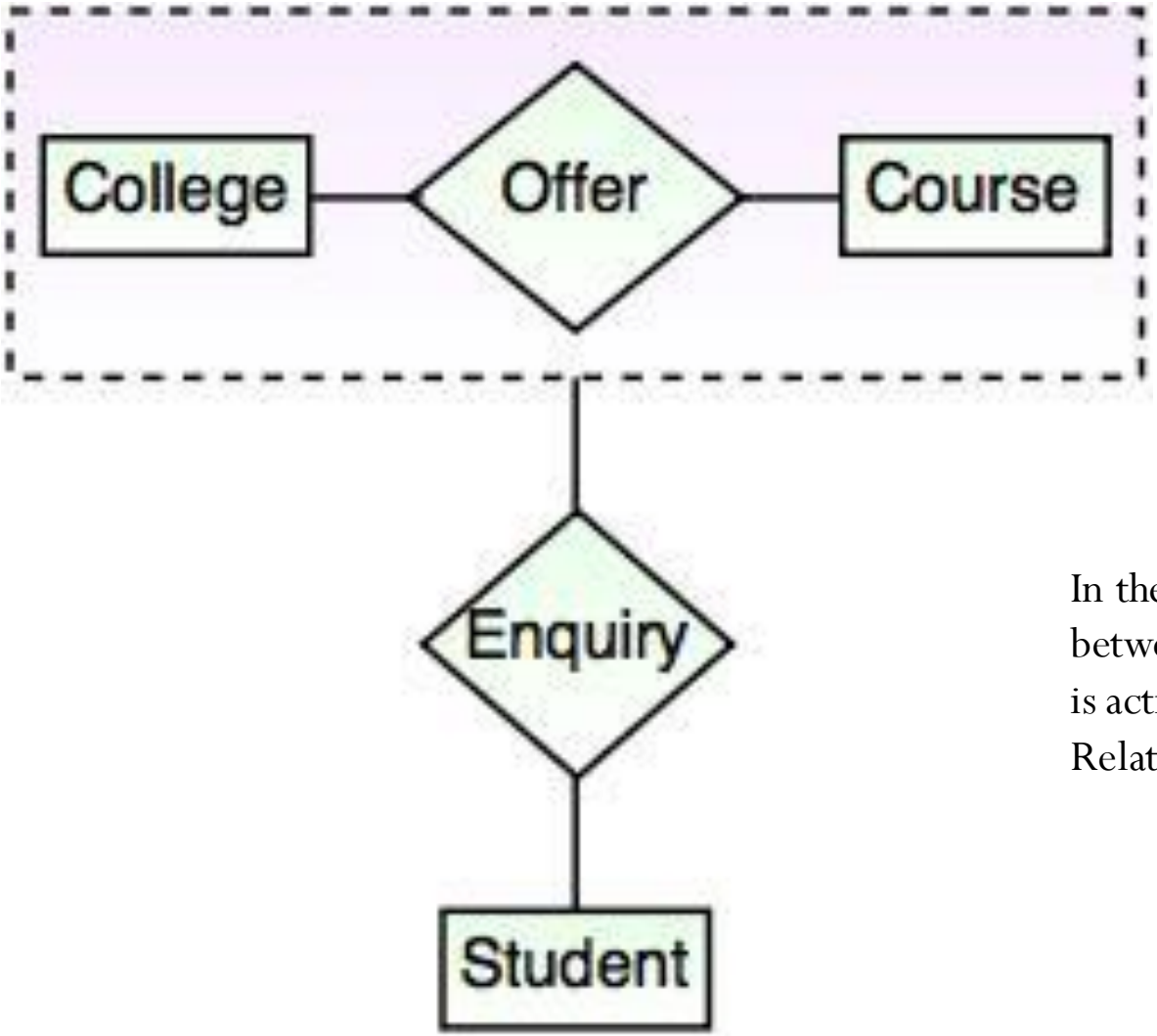
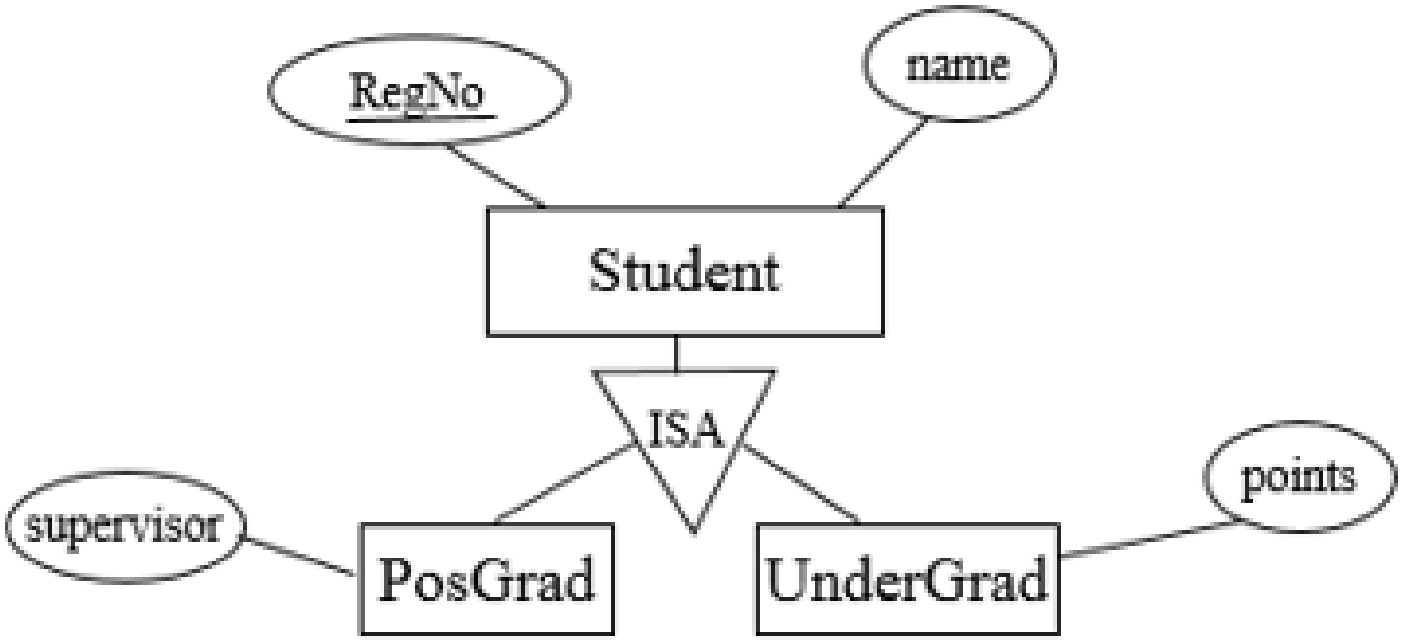


Fig. Aggregation

In the example, the relation between College and Course is acting as an Entity in Relation with Student.

Mapping specialization/generalization to relational tables

- Specialization/generalization relationship can be mapped to relational tables in three methods.
- To demonstrate the methods, we will take the student, postgraduate and undergraduate relationship.
- A student in the university has a registration number and a name.
- Only postgraduate students have supervisors. Undergraduates accumulates points through their coursework.



Method 1

All the entities in the relationship are mapped to individual tables.

Student (*Regno*, name)

PosGrad (*Regno*, supervisor)

UnderGrad (*Regno*, points)

Method 2

Only subclasses are mapped to tables. The attributes in the superclass are duplicated in all subclasses.

PosGrad (*Regno*, name, supervisor)

UnderGrad (*Regno*, name, points)

Method 3

Only the superclass is mapped to a table. The attributes in the subclasses are taken to the superclass.

Student (Regno, name, supervisor, points)

Features of EER Model

- EER creates a design more accurate to database schemas.
- It reflects the data properties and constraints more precisely.
- It includes all modeling concepts of the ER model.
- Diagrammatic technique helps for displaying the EER schema.
- It includes the concept of specialization and generalization.
- It is used to represent a collection of objects that is union of objects of different of different entity types.

Theory Digital Assignment 1 and 2

- Although you always wanted to be an artist, you ended up being an expert on databases. Your old love is still there, however, so you set up a database company, Art Base that builds a product for art galleries. The core of this product is a database with a schema that captures all the information that galleries need to maintain. Galleries keep information about artists, their names (which are unique), birthplaces, age, and style of art. For each piece of artwork, the artist, the year it was made, its unique title, its type of art (e.g., painting, lithograph, sculpture, photograph), and its price must be stored. Pieces of artwork are also classified into groups of various kinds, for example, portraits, still lifes, works by Picasso, or works of the 19th century; a given piece may belong to more than one group. Each group is identified by a name (like those just given) that describes the group. Finally, galleries keep information about customers. For each customer, galleries keep that person's unique name, address, total amount of dollars spent in the gallery (very important!), and the artists and groups of art that the customer tends to like.
- Draw the ER diagram for the database and apply Extended ER digarams features whenever possible
- Convert the Drawn ER diagram into Relational Schema