Chapter 13
Enhanced Entity-Relationship Modeling

Enhanced Entity-Relationship Model

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

Enhanced Entity-Relationship Model (cont'd)

- Most useful additional data modeling concept of Enhanced ER (EER) model is called **specialization / generalization**.
- Specialization / generalization is associated with special types of entities known as **superclasses** and **subclasses**, and the process of **attribute inheritance**.

Superclasses & subclasses

- Superclass
  - An entity type that includes one or more distinct subgroupings of its occurrences.
- Subclass
  - A distinct subgrouping of occurrences of an entity type.

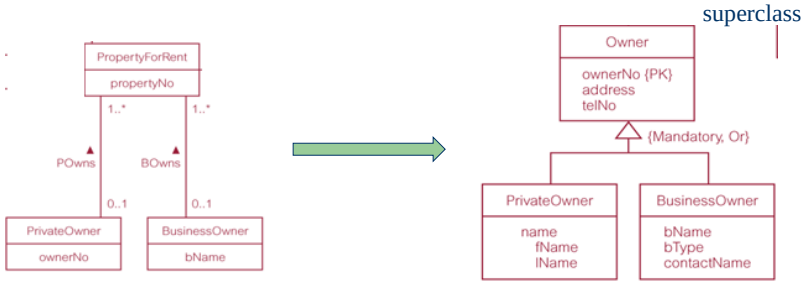
Diagrammatic Representation of specialization / generalization

- Superclass and subclass, being entities, are represented as rectangles.
- Subclasses are attached by lines to a triangle that points toward the superclass.
- The label below the specialization / generalization triangle describes the constraints on the relationship between the superclass and subclass.
- Attributes common to all subclasses are listed in the superclass.

5

Generalization Process

- Process of **identifying their common characteristics** among entities.
- A bottom-up approach that results in the identification of a **generalized superclass** from the original entity types.



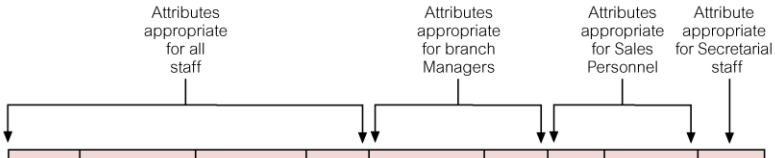
6

Specialization Process

- Process of **maximizing differences** between members of an entity by identifying their distinguishing characteristics.
- A top-down approach to defining a set of superclasses first, then by defining differences between members of an entity, related **subclasses** are identified.

Figure 13.1 AllStaff relation holding details of all staff

- Such a representation may result in a lot of nulls for the job-specific attributes.

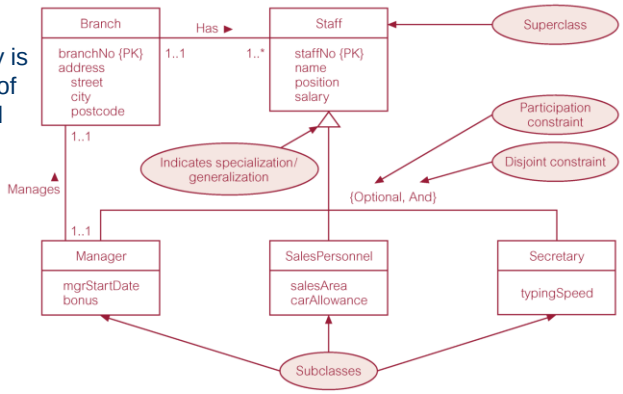


staffNo	name	position	salary	mgrStartDate	bonus	sales Area	car Allowance	typing Speed
SL21	John White	Manager	30000	01/02/95	2000			
SG37	Ann Beech	Assistant	12000					
SG66	Mary Martinez	Sales Manager	27000			SA1A	5000	
SA9	Mary Howe	Assistant	9000					
SL89	Stuart Stern	Secretary	8500					100
SL31	Robert Chin	Snr Sales Asst	17000			SA2B	3700	
SG5	Susan Brand	Manager	24000	01/06/91	2350			

Specialization of Staff entity into subclasses representing job roles

For example, members of the Staff entity type may be classified as Manager and SalesPersonnel.

In other words, the Staff entity is referred to as the superclass of the Manager, SalesPersonnel subclasses.



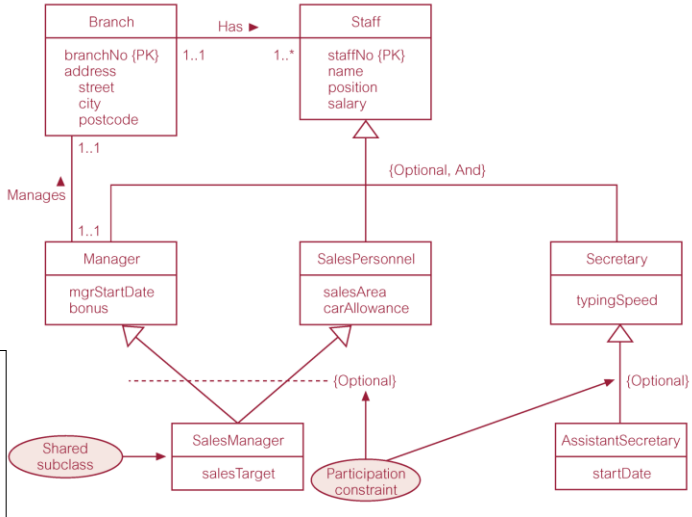
Attribute Inheritance

- We use superclasses and subclasses to avoid describing different types of staff with possibly different attributes within a single entity.
- An entity in a subclass represents same ‘real world’ object as in superclass, and may possess subclass-specific attributes, as well as those associated with the superclass.

Shared subclass

- A subclass may also have one or more subclasses.
- A subclass with more than one superclass is called a **shared subclass**. That is, a member of a shared subclass must be a member of the associated superclasses.
- An entity and its subclasses, and their subclasses, and so on, is called a **type hierarchy**.

EER diagram with shared subclass and subclass with its own subclass



A member of the SalesManager shared subclass must be a member of the SalesPersonnel and Manager subclasses as well as the Staff superclass.

Participation Constraint

- Two constraints that may apply to a specialization / generalization:
  - participation constraints;
  - disjoint constraints.
- **Participation constraint**
  - Determines whether every member in superclass must participate as a member of a subclass.
    - If true: Mandatory
    - If false: Optional
  - Represented as **{Mandatory}** and **{Optional}** respectively.

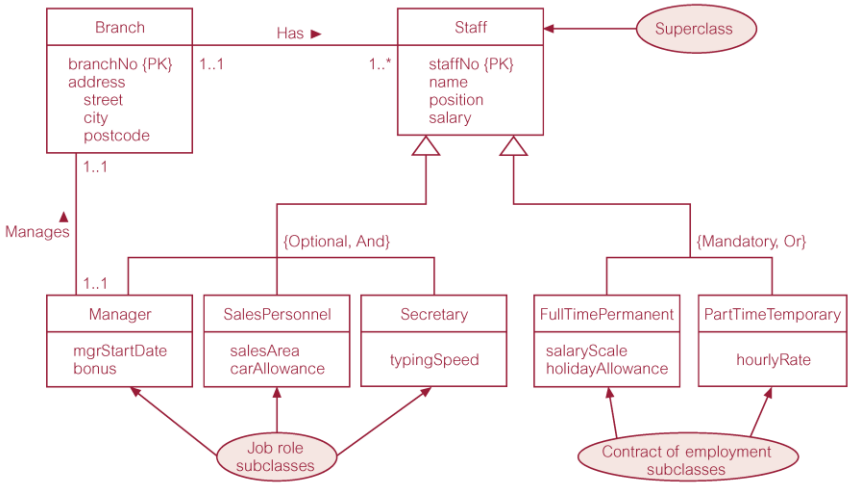
Disjoint Constraint

- Disjoint constraint
  - Describes relationship between members of the subclasses as whether overlapping exists
  - Indicates whether member of a superclass can be a member of one **{Or}** , or more than one, subclass **{And}.**
  - Applies only when a superclass has more than one subclass.

Four categories of constraints of specialization and generalization

- Mandatory and disjoint {Mandatory, Or}
- Optional and disjoint {Optional, Or}
- Mandatory and nondisjoint {Mandatory, And}
- Optional and nondisjoint {Optional, And}

Specialization/generalization of Staff entity into job roles and contracts of employment



Summary

The ER design steps are as follows:

- Identify entity types.
- Identify relationship type and the degree of the relationship type.
- Identify and associate attributes with entity or relationship types.
- Determine candidate and primary key attributes.
- Identify strong and weak entity types.
- Determine the multiplicity constraints for the relationships.
- Specialize/generalize entity types.
- Draw ER diagram.