Database Design and Practice 1
Lecture 03

Enhanced Entity-Relationship Modelling & conceptual data modelling methodology

Lecture 03 - Objectives

- Limitations of basic concepts of the ER model and requirements to represent more complex applications.
- Specialization/generalization: useful additional data modelling concept of EER model.
- Summary of step-by-step approach to use ER modelling to develop a conceptual data model
- Validation of validity and accuracy of conceptual model
- Documentation of the conceptual design process

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.

The Enhanced Entity-Relationship Model

- **Semantic concepts are incorporated into the original ER model and called the Enhanced Entity-Relationship (EER) model.**

- **Examples of additional concept of EER model is called specialization / generalization.**

Specialization / Generalization

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

Specialization / Generalization

- Superclass/subclass relationship is one-to-one (1:1).
- Superclass may contain overlapping or distinct subclasses.
- Not all members of a superclass need be a member of a subclass.

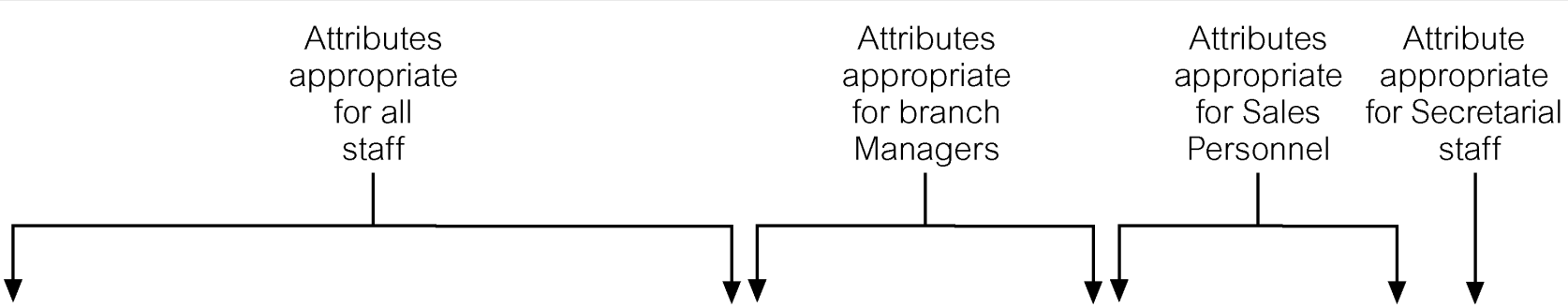
Specialization / Generalization

- **Attribute Inheritance**
  - **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.**

Specialization / Generalization

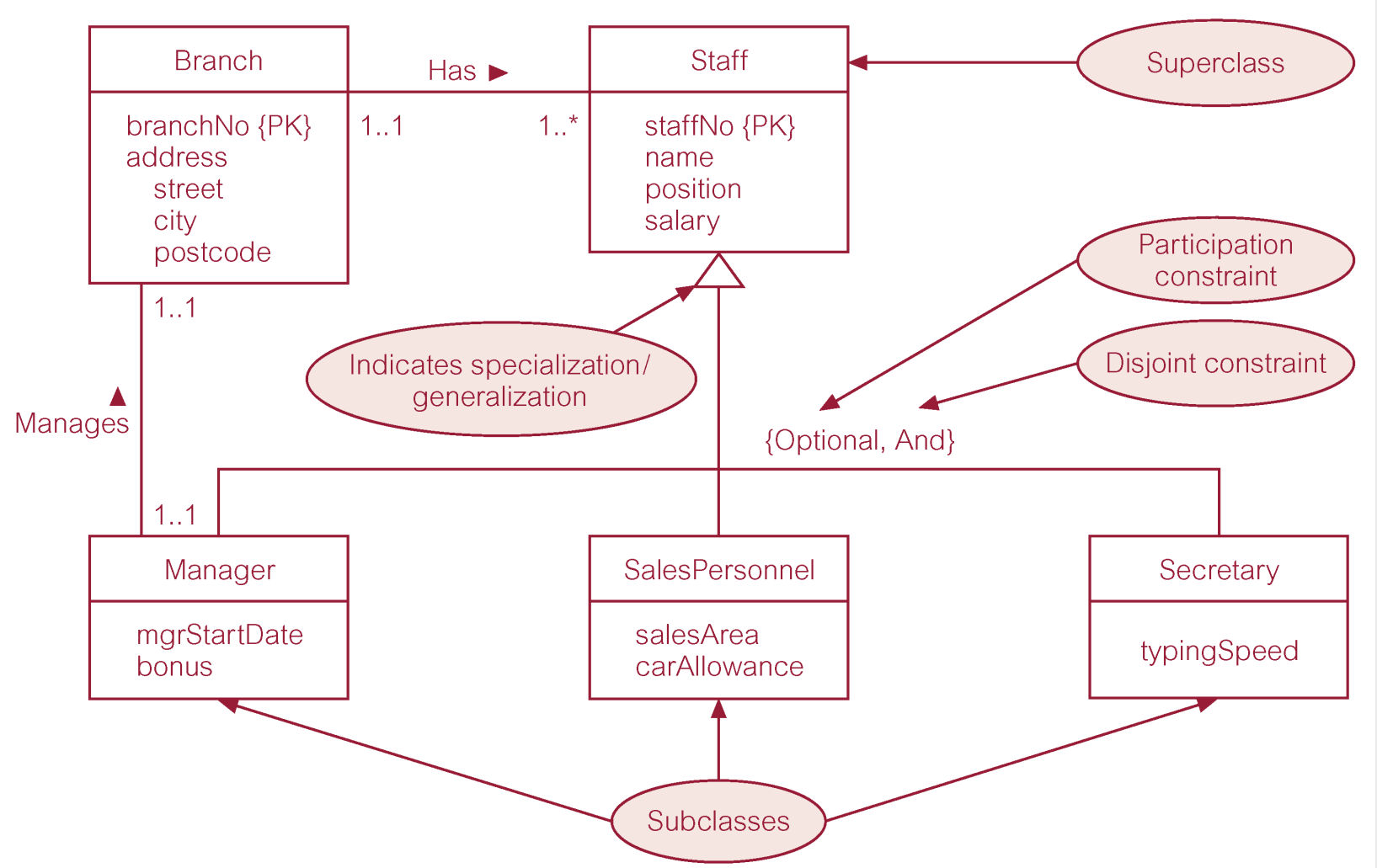
- **Specialization**
  - **Process of maximizing differences between members of an entity by identifying their distinguishing characteristics.**
- **Generalization**
  - **Process of minimizing differences between entities by identifying their common characteristics.**

AllStaff relation holding details of all staff

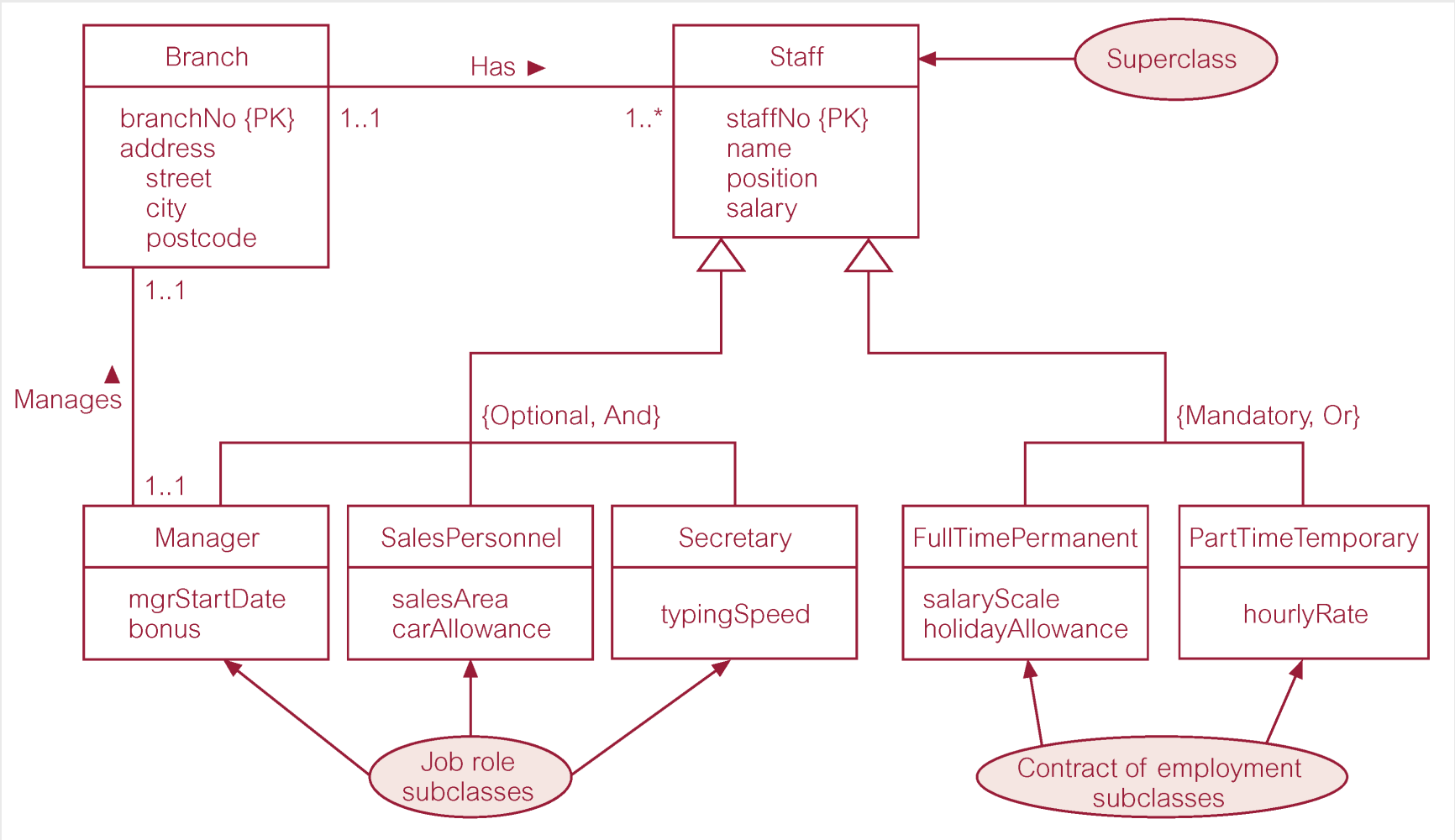


staffNo	name	position	salary	mgrStartDate	bonus	sales Area	car Allowance	typing Speed
SL21	John White	Manager	30000	01/02/95	2000			
SG37	Ann Beech	Assistant	12000					
SG66	Mary Martinez	Sales Manager	27000			SA1A	5000	
SA9	Mary Howe	Assistant	9000					
SL89	Stuart Stern	Secretary	8500					100
SL31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

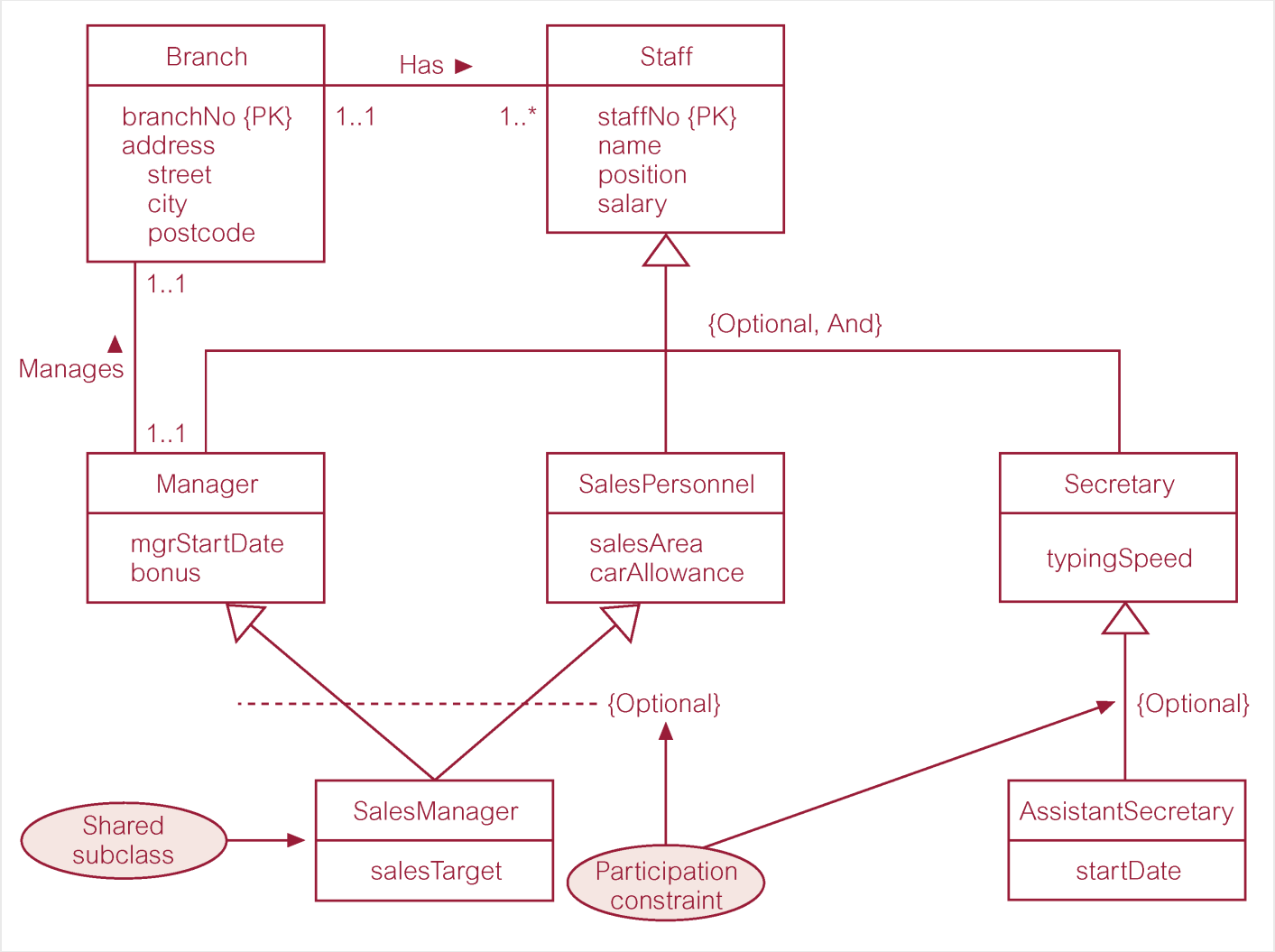
Specialization/generalization of Staff entity into subclasses representing job roles



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



EER diagram with shared subclass and subclass with its own subclass



Constraints on Specialization / Generalization

- **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.**
  - **May be *mandatory* or *optional*.**

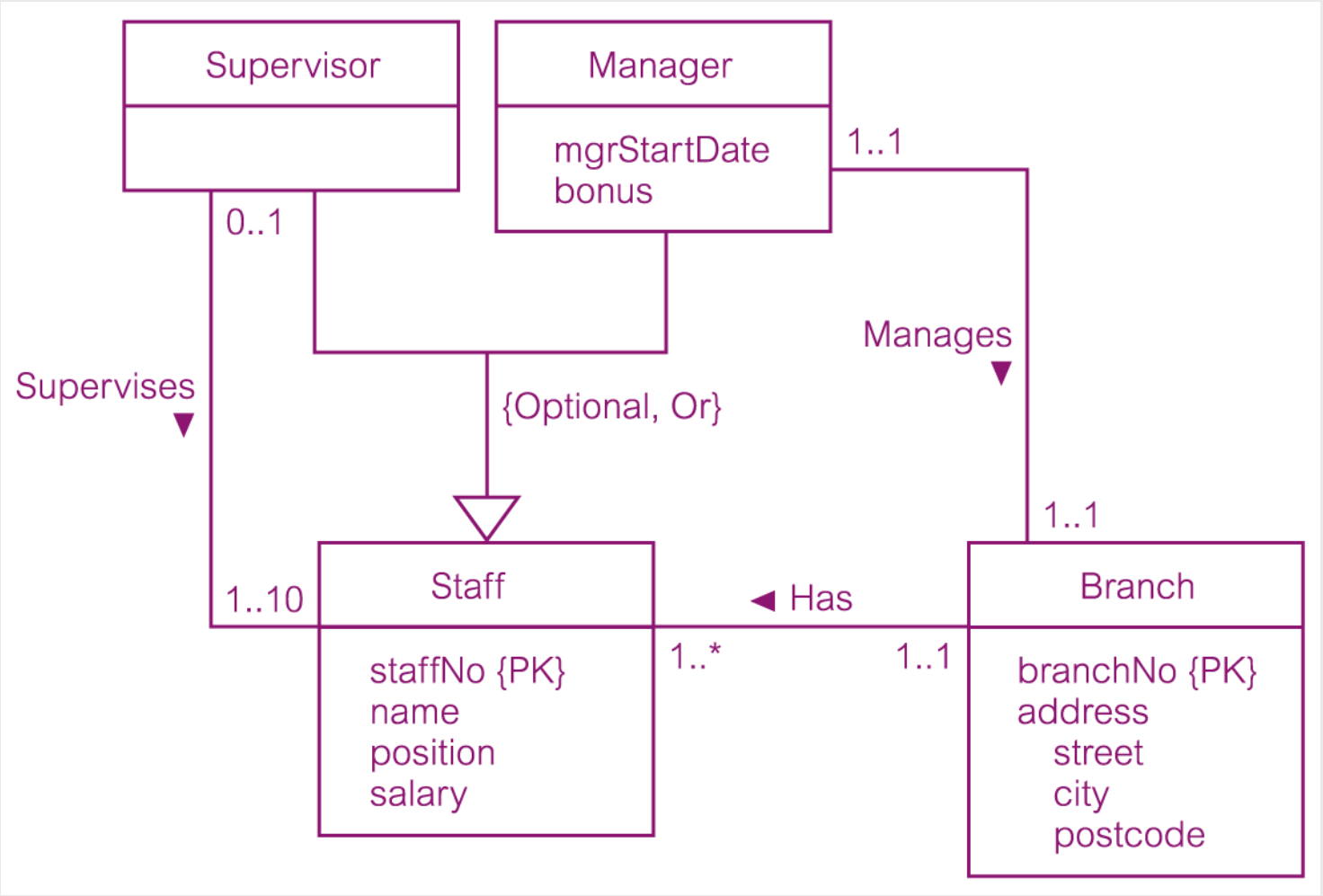
Constraints on Specialization / Generalization

- **Disjoint constraint**
  - **Describes relationship between members of the subclasses and indicates whether member of a superclass can be a member of one, or more than one, subclass.**
  - **May be *disjoint* or *nondisjoint*.**

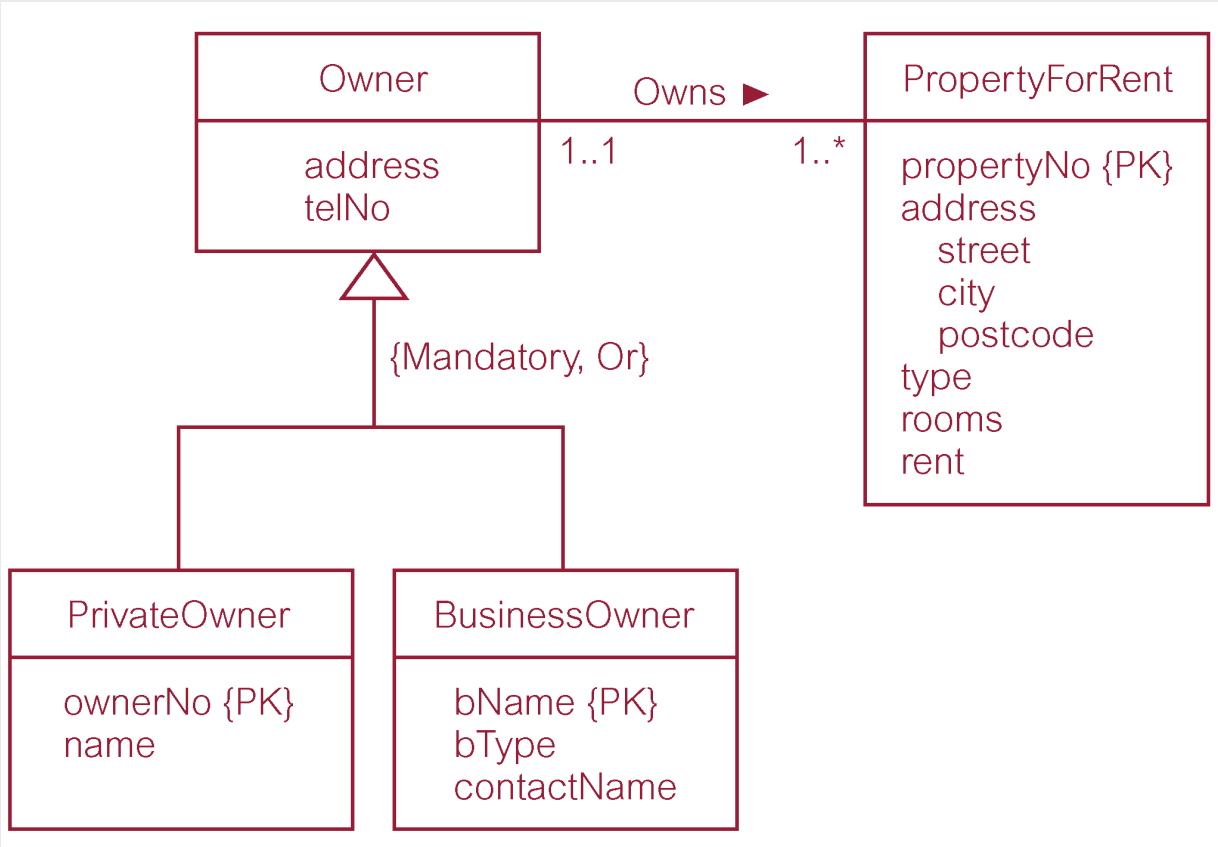
Constraints on Specialization / Generalization

- There are four categories of constraints of specialization and generalization:
  - mandatory and disjoint
  - optional and disjoint
  - mandatory and nondisjoint
  - optional and nondisjoint.

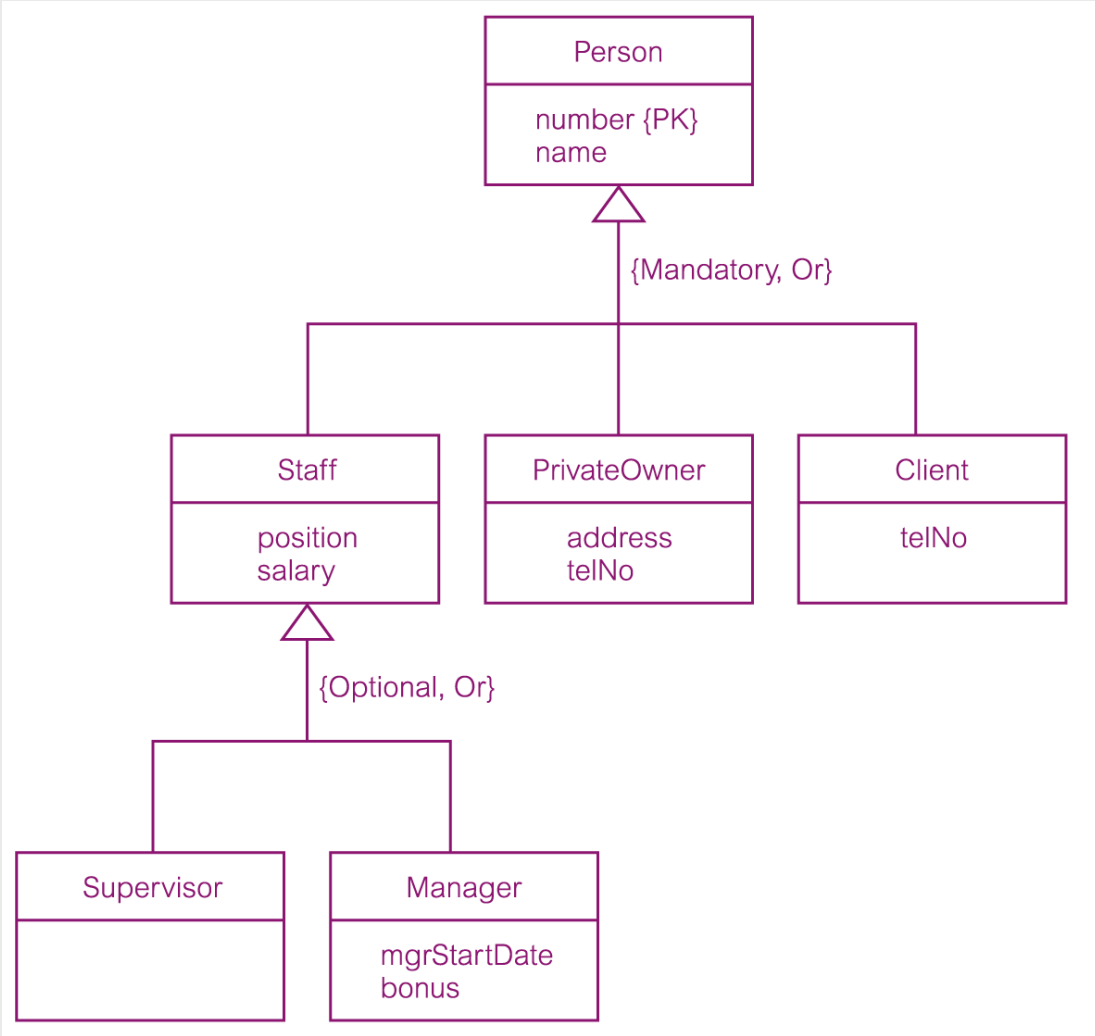
DreamHome worked example - Staff Superclass with Supervisor and Manager subclasses



DreamHome worked example - Owner Superclass with PrivateOwner and BusinessOwner subclasses



DreamHome worked example - Person superclass with Staff, PrivateOwner, and Client subclasses



Conceptual Database Design Methodology

- Step 1 Build conceptual data model
  - Step 1.1 Identify entity types
  - Step 1.2 Identify relationship types
  - Step 1.3 Identify and associate attributes with entity or relationship types
  - Step 1.4 Determine attribute domains
  - Step 1.5 Determine candidate, primary, and alternate key attributes

Conceptual Database Design Methodology (cont)

- Step 1 Build conceptual data model (continue)
  - Step 1.6 Consider use of enhanced modeling concepts (optional step)
  - Step 1.7 Check model for redundancy
  - Step 1.8 Validate conceptual model against user transactions
  - Step 1.9 Review conceptual data model with user

Step 1 Build Conceptual Data Model

- To build a conceptual data model of the data requirements of the enterprise.
  - Model comprises entity types, relationship types, attributes and attribute domains, primary and alternate keys, and integrity constraints.

- Step 1.1 Identify entity types
  - To identify the required entity types.

- Step 1.2 Identify relationship types
  - To identify the important relationships that exist between the entity types.

Step 1 Build Conceptual Data Model

- **Step 1.3 Identify and associate attributes with entity or relationship types**
  - To associate attributes with the appropriate entity or relationship types and document the details of each attribute.
- **Step 1.4 Determine attribute domains**
  - To determine domains for the attributes in the data model and document the details of each domain.

Step 1 Build Conceptual Data Model

- **Step 1.5 Determine candidate, primary, and alternate key attributes**
  - To identify the candidate key(s) for each entity and if there is more than one candidate key, to choose one to be the primary key and the others as alternate keys.
- **Step 1.6 Consider use of enhanced modeling concepts (optional step)**
  - To consider the use of enhanced modeling concepts, such as specialization / generalization, aggregation, and composition.

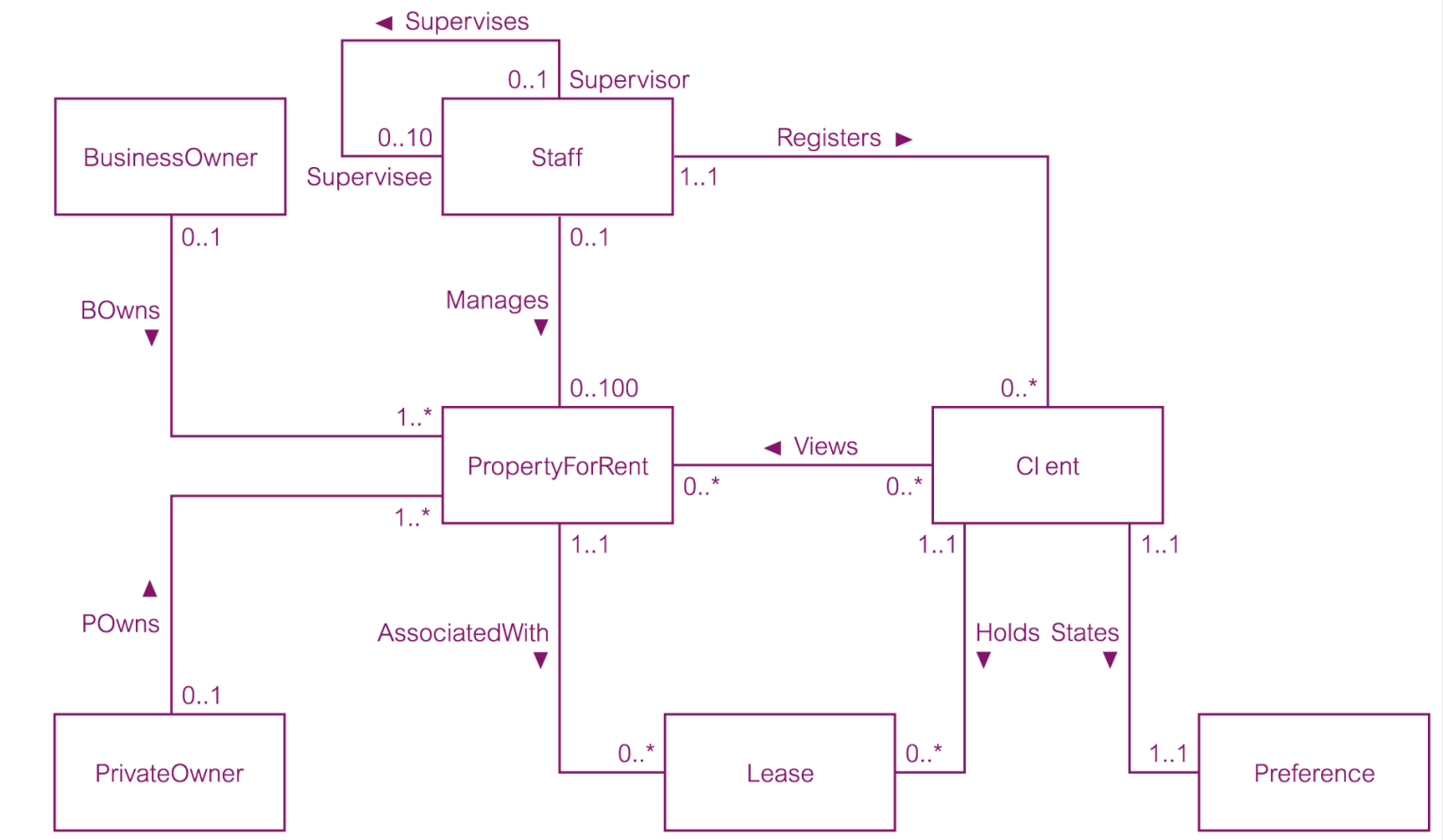
Step 1 Build Conceptual Data Model

- **Step 1.7 Check model for redundancy**
  - To check for the presence of any redundancy in the model and to remove any that does exist.
- **Step 1.8 Validate conceptual model against user transactions**
  - To ensure that the conceptual model supports the required transactions.
- **Step1.9 Review conceptual data model with user**
  - To review the conceptual data model with the user to ensure that the model is a 'true' representation of the data requirements of the enterprise.

Extract from data dictionary for Staff user views of *DreamHome* showing description of entities

Entity name	*Description*	*Aliases*	*Occurrence*
Staff	General term descr bing al staff employed by *DreamHome*.	Employee	Each member of staff works at one particular branch.
PropertyForRent	General term descr bing all property for rent.	Property	Each property has a s ngle owner and is available at one specific branch, where the property is managed by one member of staff. A property is viewed by many clients and rented by a single client, at any one time.

First-cut ER diagram for Staff user views of *DreamHome*



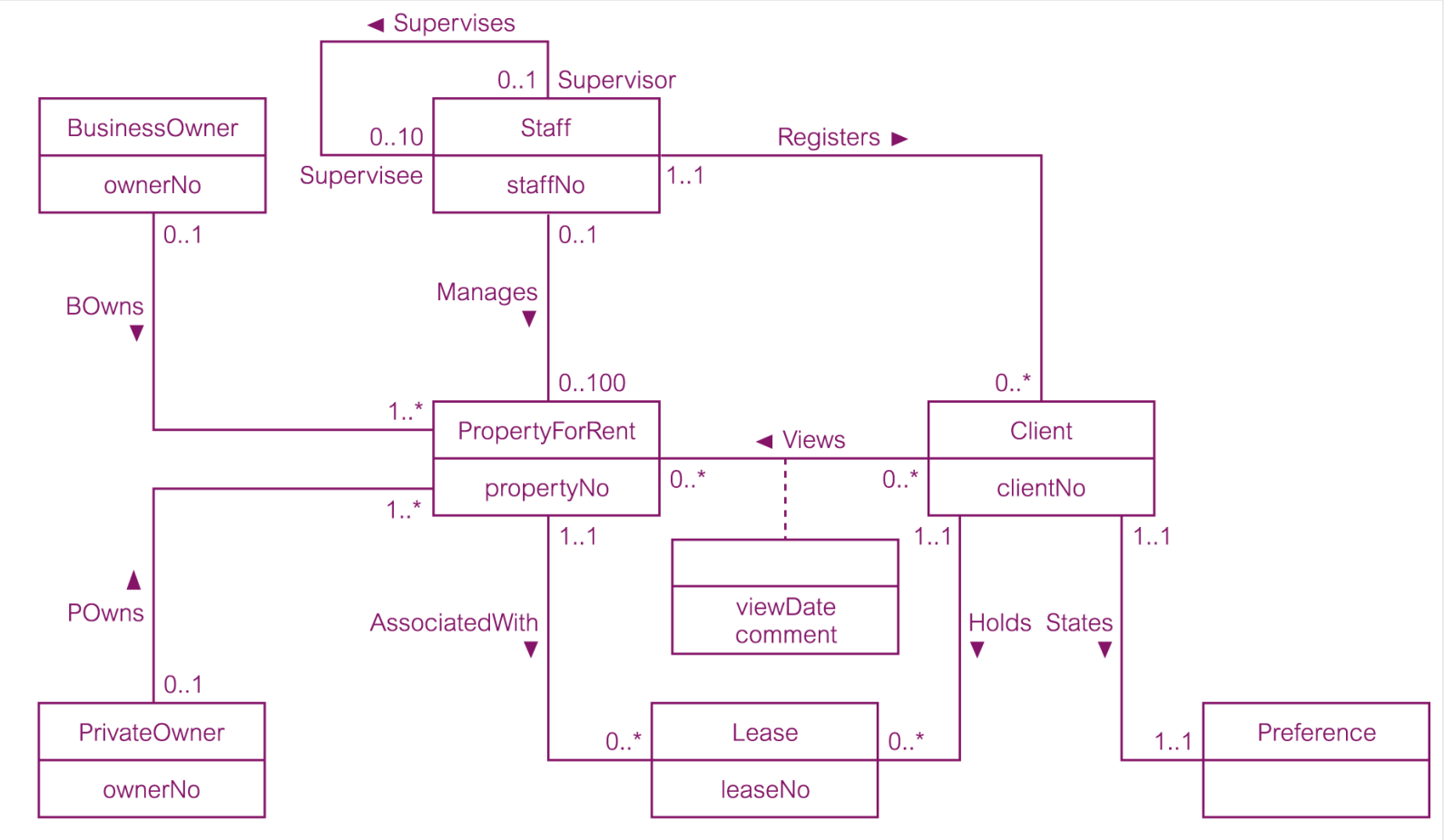
Extract from data dictionary for Staff user views of *DreamHome* showing description of relationships

Entity name	*Multiplicity*	*Relationship*	*Multiplicity*	*Entity name*
Staff	0..1 0..1	*Manages* *Supervises*	0..100 0..10	**PropertyForRent** **Staff**
PropertyForRent	1..1	*AssociatedWith*	0..*	**Lease**

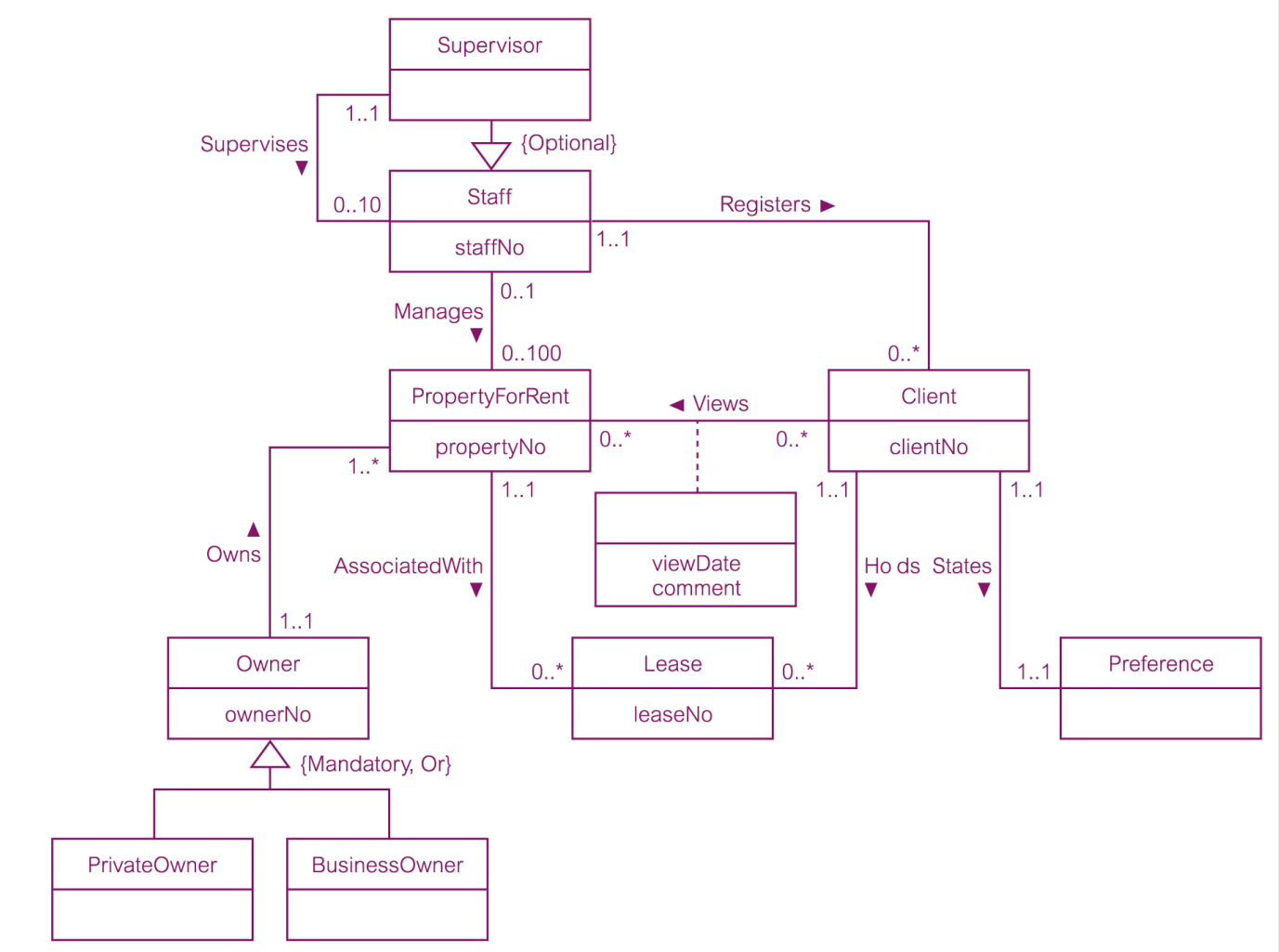
Extract from data dictionary for Staff user views of *DreamHome* showing description of attributes

Entity name	*Attributes*	*Description*	*Data Type & Length*	*Nulls*	*Multi-valued*	. . .
Staff	**staffNo**	Unique y identifies a member of staff	5 variable characters	No	No	
	name					
	fName	First name of staff	15 variable characters	No	No	
	lName	Last name of staff	15 variable characters	No	No	
	position	Job title of member of staff	10 variable characters	No	No	
	sex	Gender of member of staff	1 character (M or F)	Yes	No	
	DOB	Date of birth of member of staff	Date	Yes	No	
PropertyForRent	**propertyNo**	Unique y identifies a property for rent	5 variable characters	No	No	

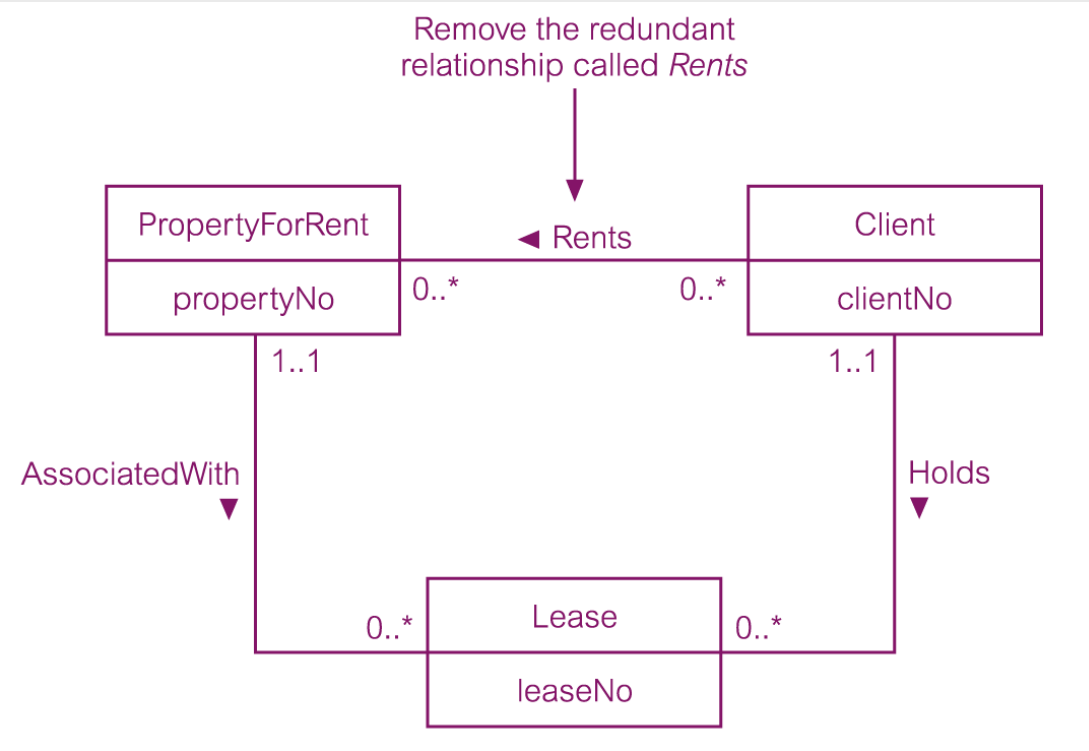
ER diagram for Staff user views of *DreamHome* with primary keys added



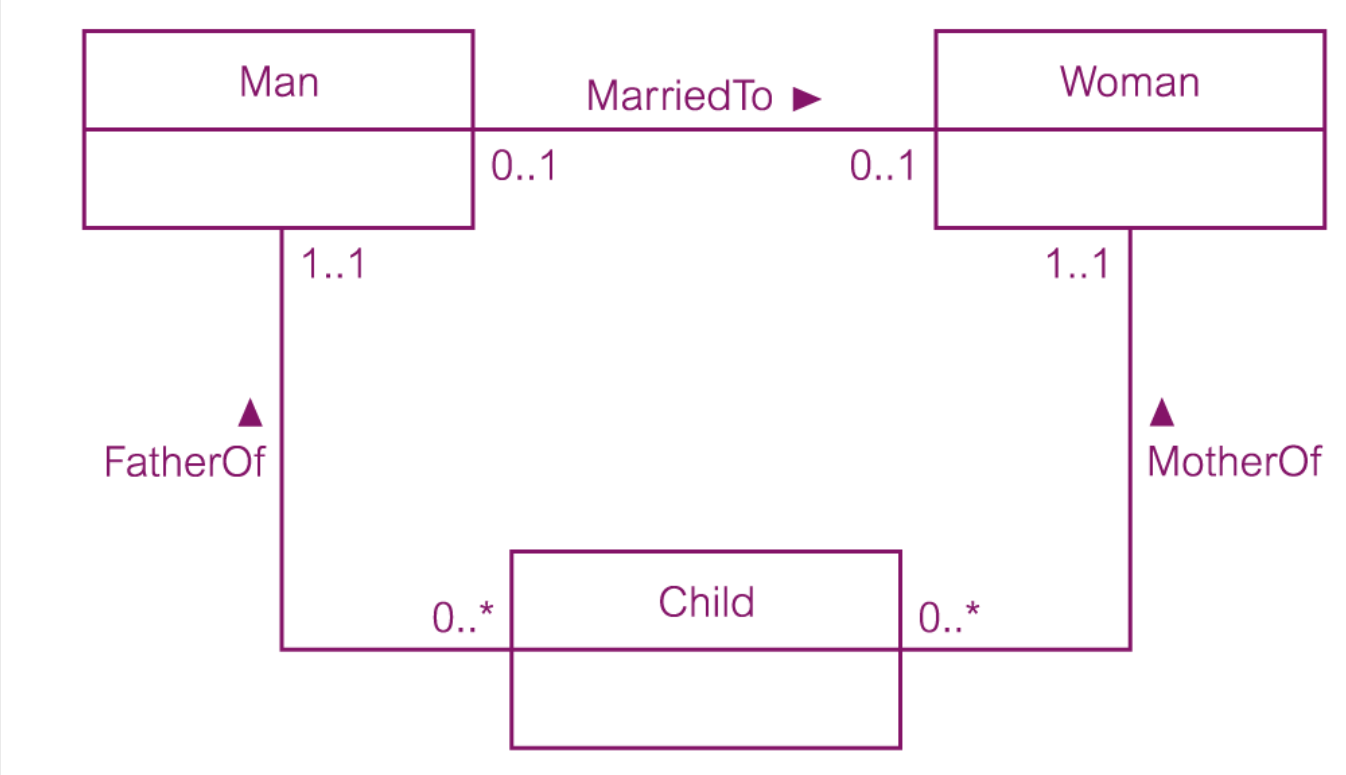
Revised ER diagram for Staff user views of *DreamHome* with specialization / generalization



Example of removing a redundant relationship called *Rents*



Example of a non-redundant relationship *FatherOf*



Using pathways to check that the conceptual model supports the user transactions