CSE 204 - INTRO TO DATABASE SYSTEMS
CONCEPTUAL DATABASE DESIGN

Joseph LEDET
Department of Computer Engineering
Akdeniz University
josephledet@akdeniz.edu.tr

OUTLINE

- The purpose of a design methodology.
- Database design has three main phases: conceptual, logical, and physical design.
- How to decompose the scope of the design into specific views of the enterprise.
- How to use Entity–Relationship (ER) modeling to build a conceptual data model based on the data requirements of an enterprise.
- How to validate the resultant conceptual model to ensure it is a true and accurate representation of the data requirements enterprise.
- How to document the process of conceptual database design.
- End-users play an integral role throughout the process of conceptual database design.

DESIGN METHODOLOGY

- A structured approach that uses procedures, techniques, tools, and documentation aids to support and facilitate the process of design.

DATABASE DESIGN METHODOLOGY

- Three main phases
  - Conceptual database design
  - Logical database design
  - Physical database design

CONCEPTUAL DATABASE DESIGN

- The process of constructing a model of the data used in an enterprise, independent of all physical considerations.

LOGICAL DATABASE DESIGN

- The process of constructing a model of the data used in an enterprise based on a specific data model (e.g. relational), but independent of a particular DBMS and other physical considerations.

PHYSICAL DATABASE DESIGN

- The process of producing a description of the implementation of the database on secondary storage; it describes the base relations, file organizations, and indexes design used to achieve efficient access to the data, and any associated integrity constraints and security measures.

CRITICAL SUCCESS FACTORS IN DATABASE DESIGN

- Work interactively with the users as much as possible.
- Follow a structured methodology throughout the data modeling process.
- Employ a data-driven approach.
- Incorporate structural and integrity considerations into the data models.
- Combine conceptualization, normalization, and transaction validation techniques into the data modeling methodology.

CRITICAL SUCCESS FACTORS IN DATABASE DESIGN

- Use diagrams to represent as much of the data models as possible.
- Use a Database Design Language (DBDL) to represent additional data semantics.
- Build a data dictionary to supplement the data model diagrams.
- Be willing to repeat steps.

OVERVIEW DATABASE DESIGN METHODOLOGY

Conceptual database design

- 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
  - 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

OVERVIEW DATABASE DESIGN METHODOLOGY

Logical database design

- Step 2 Build and validate logical data model
    - Step 2.1 Derive relations for logical data model
    - Step 2.2 Validate relations using normalization
    - Step 2.3 Validate relations against user transactions
    - Step 2.4 Define integrity constraints
    - Step 2.5 Review logical data model with user
    - Step 2.6 Merge logical data models into global model (optional step)
    - Step 2.7 Check for future growth

OVERVIEW DATABASE DESIGN METHODOLOGY

Physical database design

- Step 3 Translate logical data model for target DBMS
  - Step 3.1 Design base relations
  - Step 3.2 Design representation of derived data
  - Step 3.3 Design general constraints
- Step 4 Design file organizations and indexes
  - Step 4.1 Analyze transactions
  - Step 4.2 Choose file organization
  - Step 4.3 Choose indexes
  - Step 4.4 Estimate disk space requirements

OVERVIEW DATABASE DESIGN METHODOLOGY

- Step 5 Design user views
- Step 6 Design security mechanisms
- Step 7 Consider the introduction of controlled redundancy
- Step 8 Monitor and tune the operational system

STEP 1 BUILD CONCEPTUAL DATA

- 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.

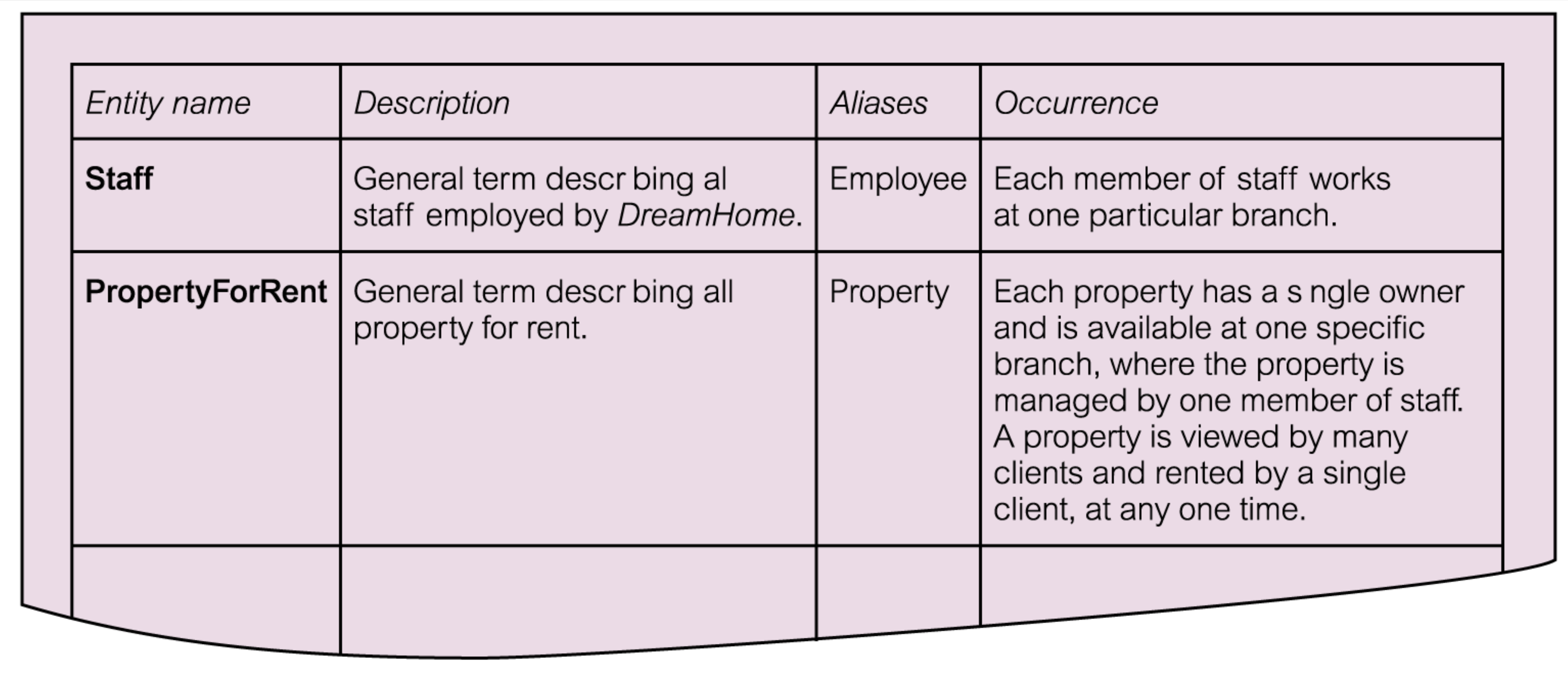
STEP 1 BUILD CONCEPTUAL DATA

- 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 1.5 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

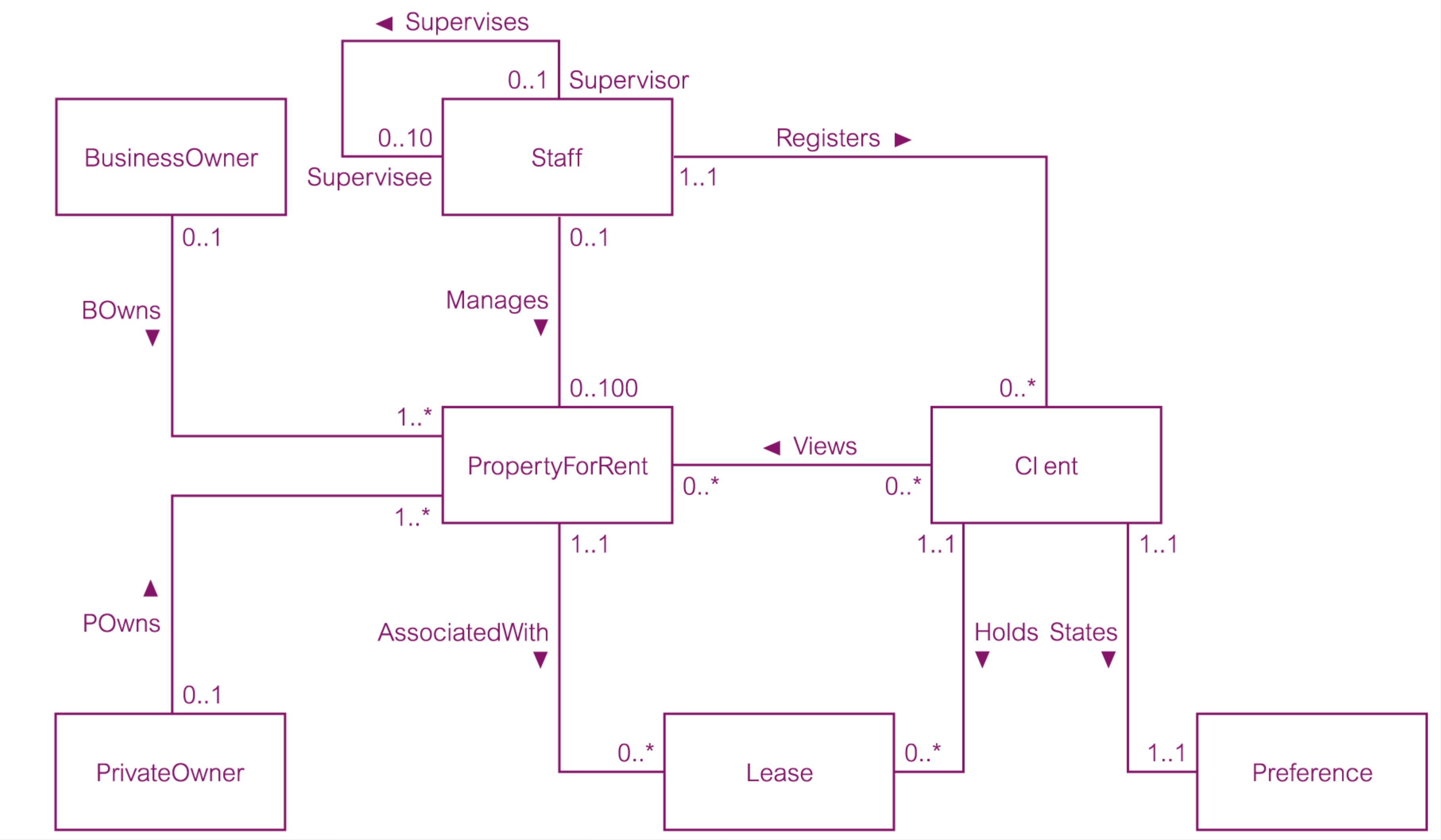
- Step 1.7 Check model for redundancy
  - To check for the presence of any redundancy in the model and to remove any that does exist.
- Step 1.8 Validate conceptual model against user transactions
  - To ensure that the conceptual model supports the required transactions.
- Step1.9 Review conceptual 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

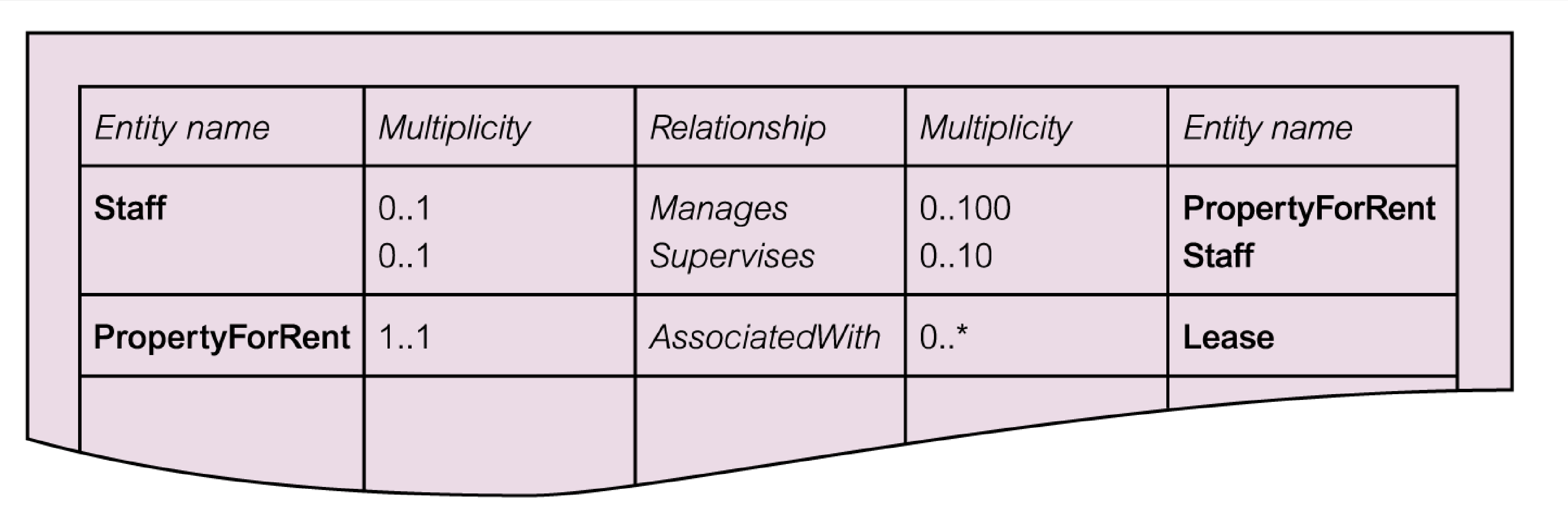


Entity name	*Description*	*Aliases*	*Occurrence*
Staff	General term descr bing al staff employed by *DreamHome*.	Employee	Each member of staff works at one particular branch.
PropertyForRent	General term descr bing all property for rent.	Property	Each property has a s ngle owner and is available at one 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

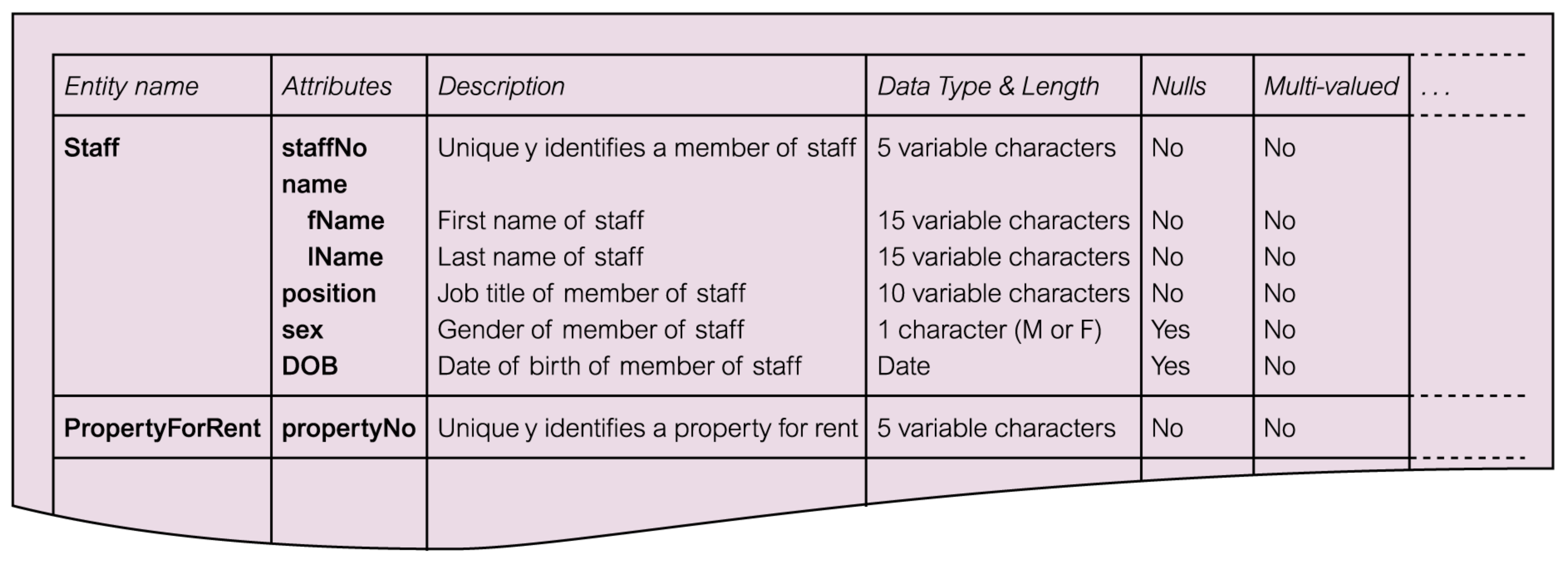


EXTRACT FROM DATA DICTIONARY FOR STAFF USER VIEWS OF DREAMHOME SHOWING DESCRIPTION OF RELATIONSHIPS



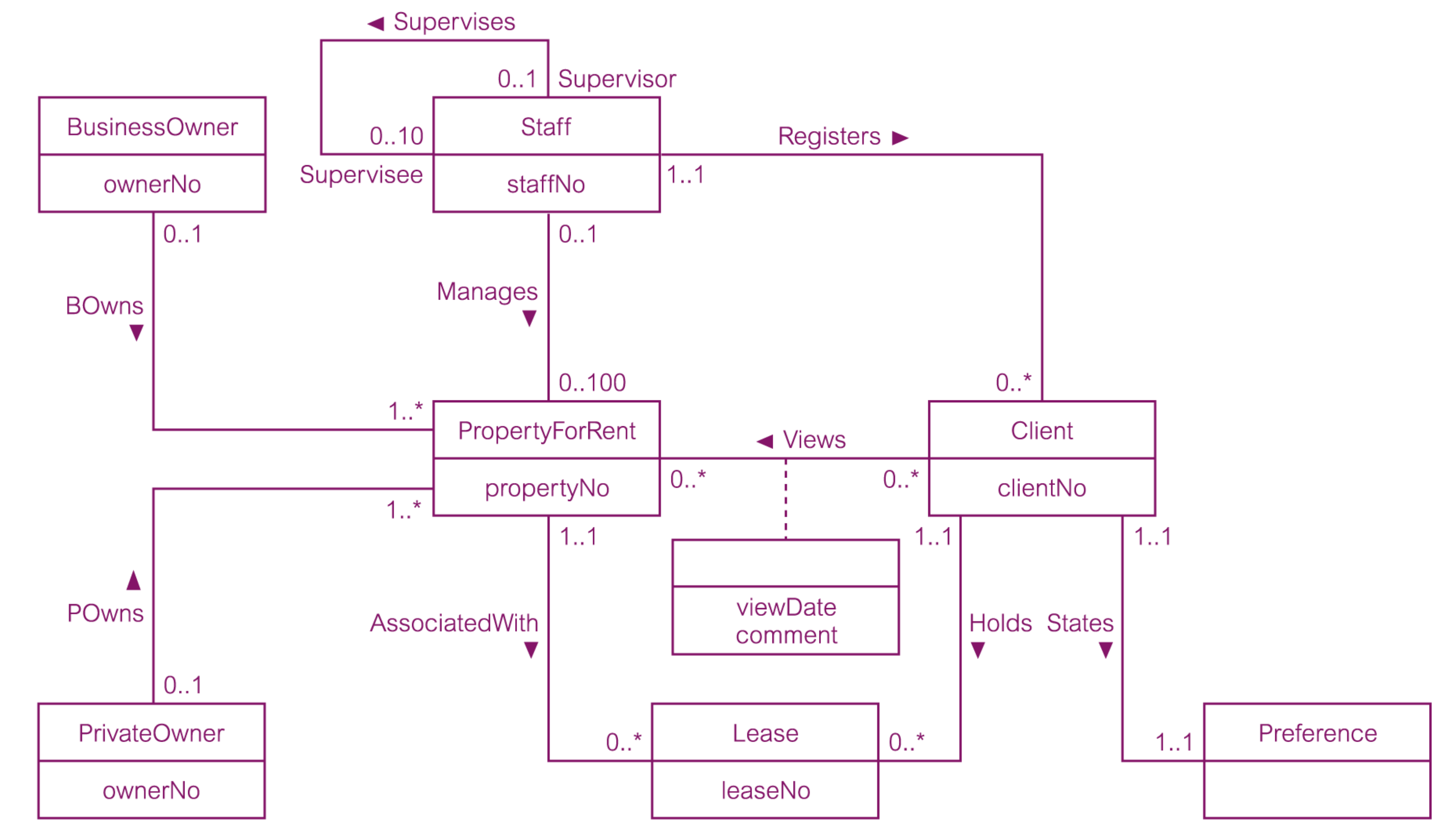
Entity name	*Multiplicity*	*Relationship*	*Multiplicity*	*Entity name*
Staff	0..1 0..1	*Manages* *Supervises*	0..100 0..10	**PropertyForRent** **Staff**
PropertyForRent	1..1	*AssociatedWith*	0..*	**Lease**

EXTRACT FROM DATA DICTIONARY FOR STAFF USER VIEWS OF DREAMHOME SHOWING DESCRIPTION OF ATTRIBUTES

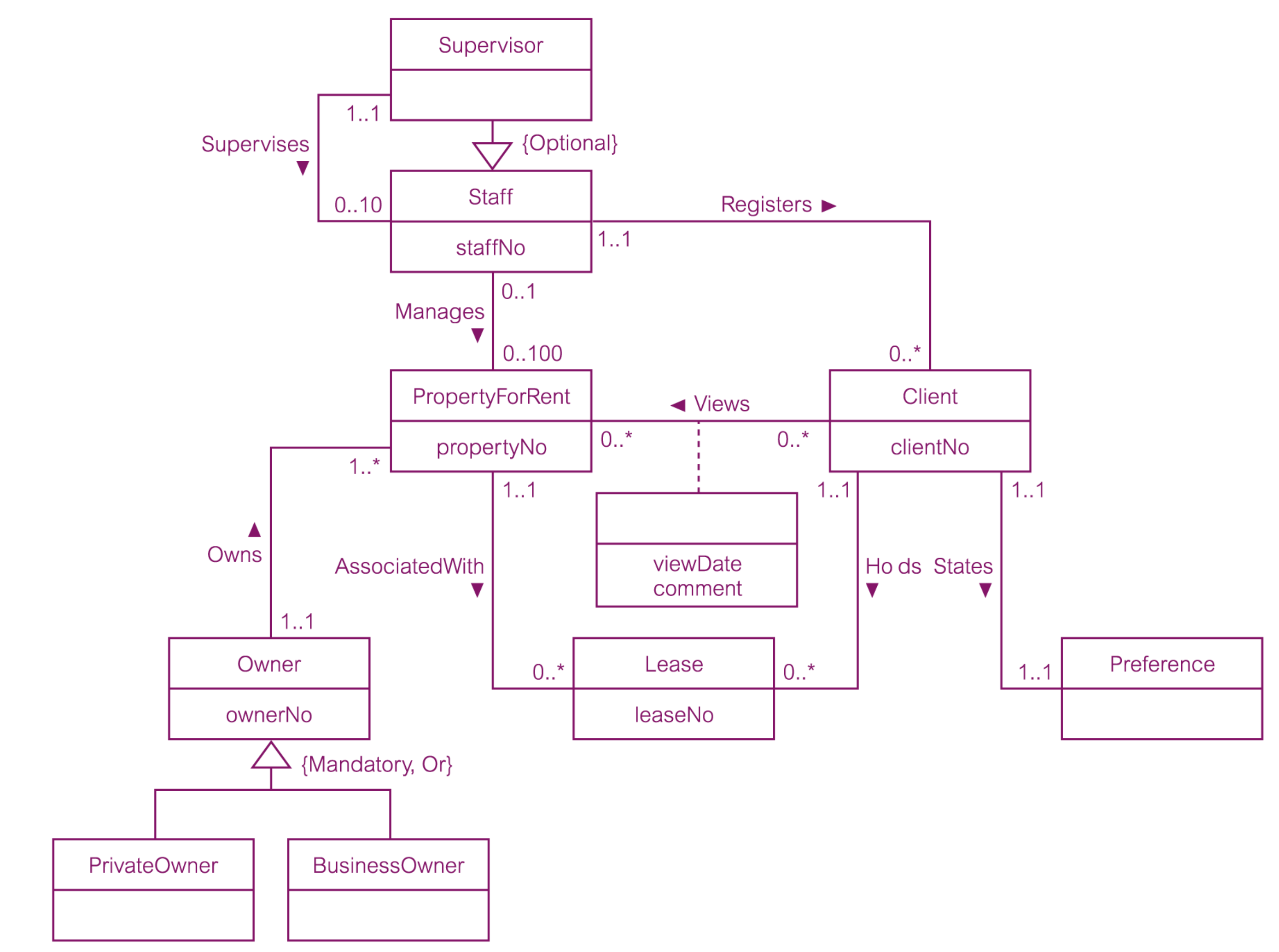


Entity name	*Attributes*	*Description*	*Data Type & Length*	*Nulls*	*Multi-valued*	. . .
Staff	**staffNo**	Unique y identifies a member of staff	5 variable characters	No	No	
	name					
	fName	First name of staff	15 variable characters	No	No	
	lName	Last name of staff	15 variable characters	No	No	
	position	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	

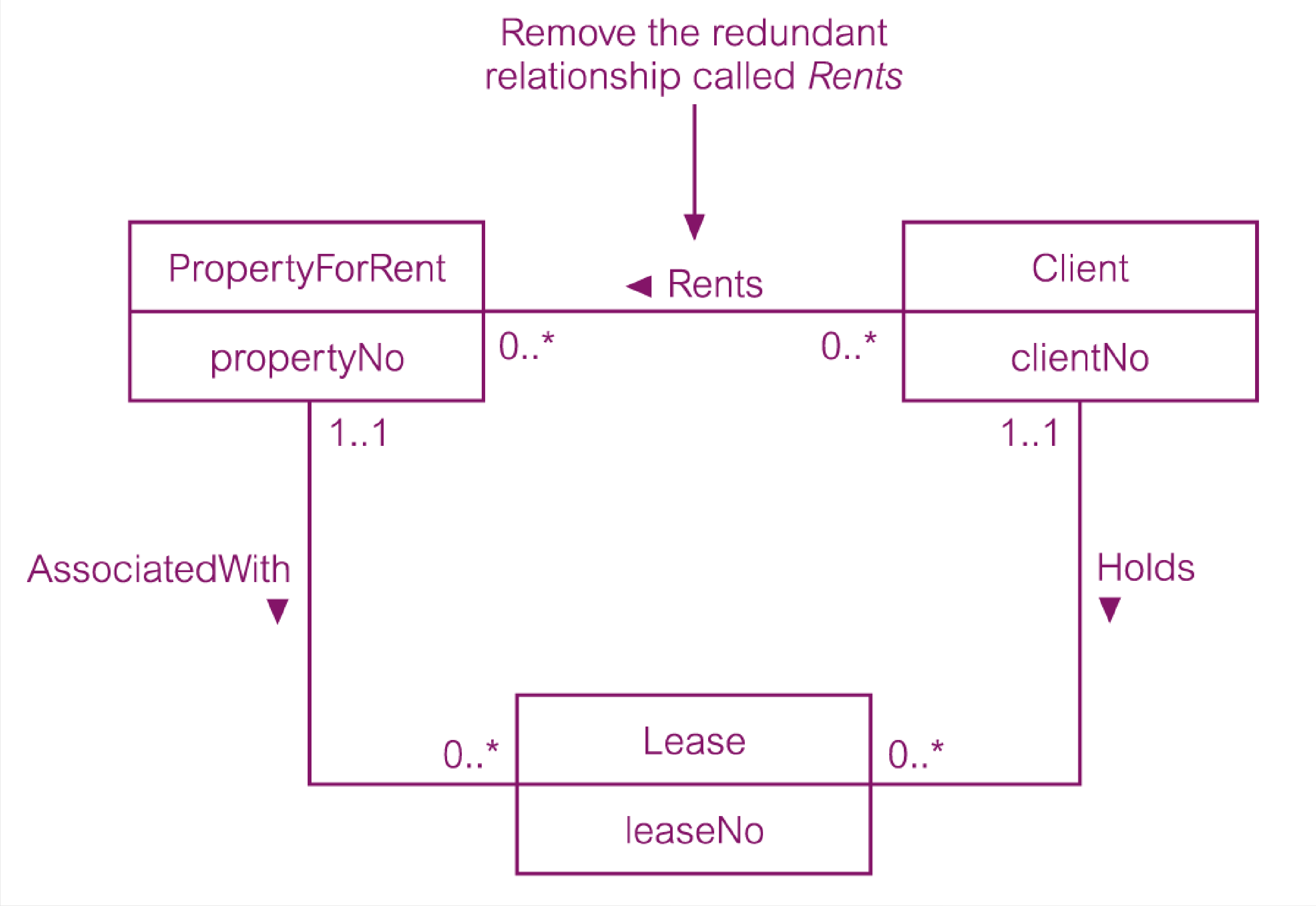
ER DIAGRAM FOR STAFF USER VIEWS OF DREAMHOME WITH PRIMARY KEYS ADDED



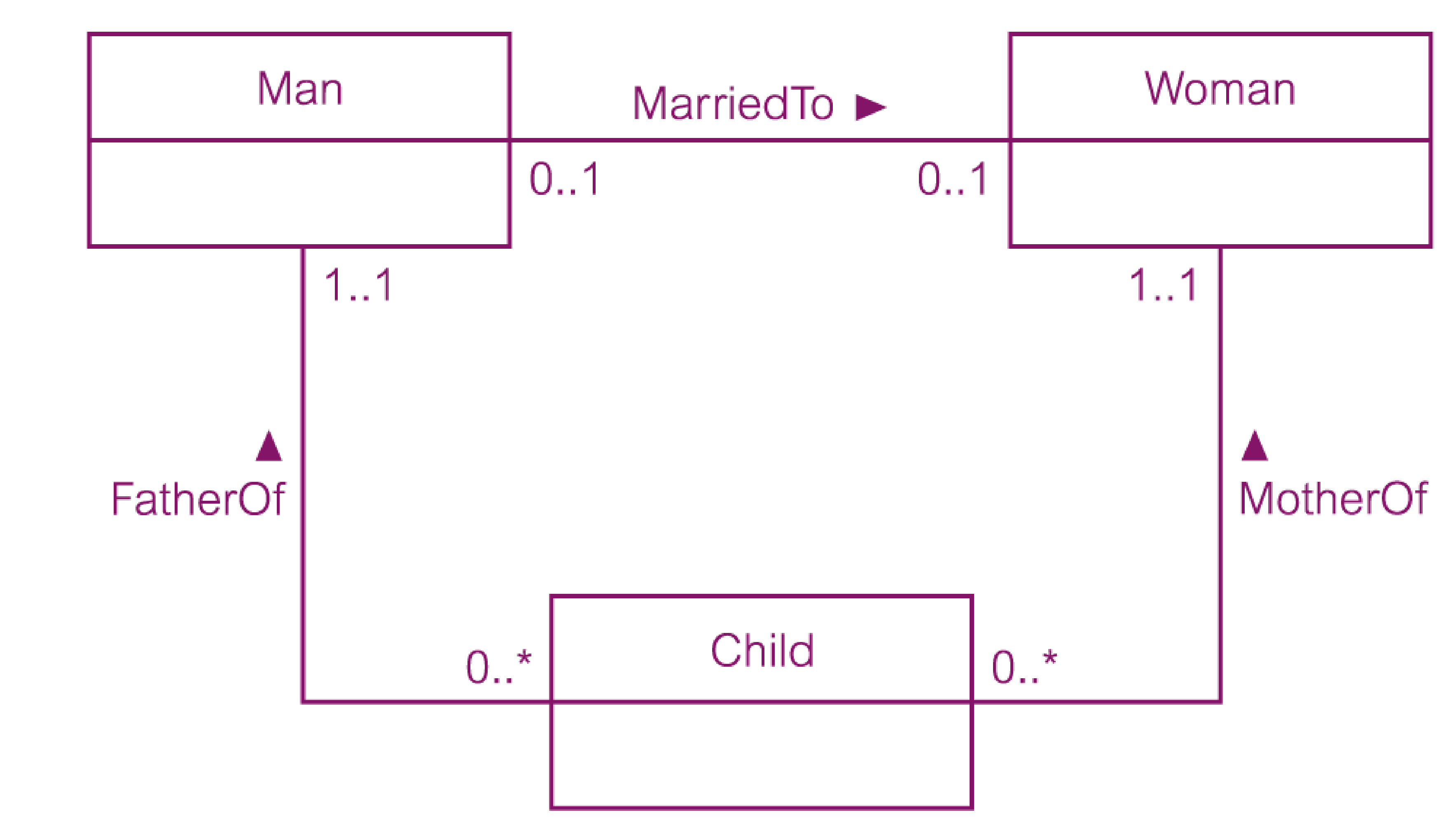
REVISED ER DIAGRAM FOR STAFF USER VIEWS OF DREAMHOME WITH SPECIALIZATION / GENERALIZATION



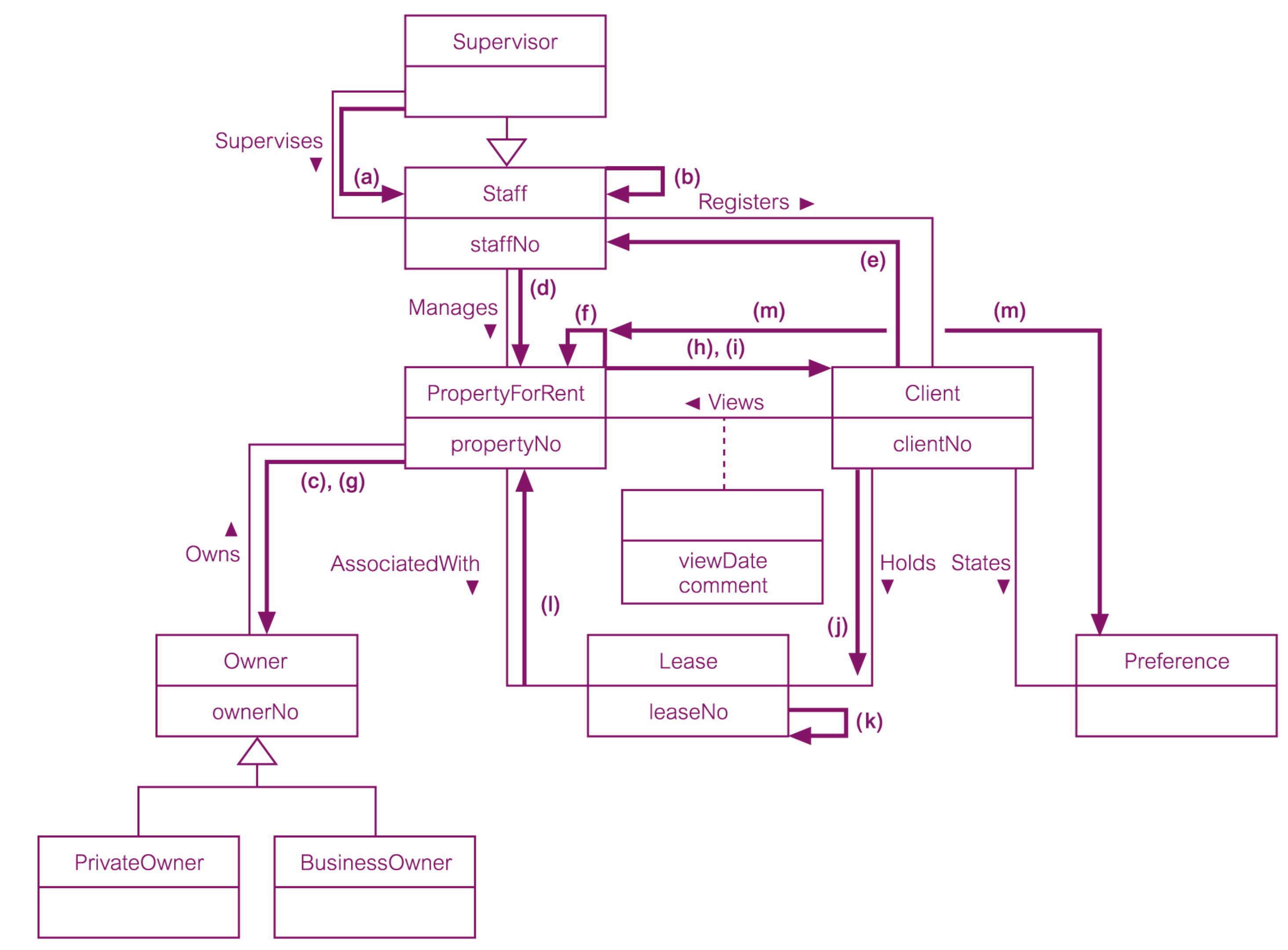
EXAMPLE OF REMOVING A REDUNDANT RELATIONSHIP CALLED RENTS



EXAMPLE OF A NON-REDUNDANT RELATIONSHIP FATHEROF



USING PATHWAYS TO CHECK THAT THE CONCEPTUAL MODEL SUPPORTS THE USER TRANSACTIONS


Supervisor
Supervises ▼
(a)
Staff
staffNo
(b)
Registers ►
(e)
Manages ▼
(d)
(f)
(m)
(h), (i)
(m)
PropertyForRent
propertyNo
◄ Views
Client
clientNo
(c), (g)
Owns ▲
AssociatedWith ▼
(l)
viewDate
comment
Holds ▼
States ▼
(j)
Owner
ownerNo
Lease
leaseNo
(k)
Preference
PrivateOwner
BusinessOwner