UNTAR
Universitas Tarumanagara

Terakreditasi BAN PT A (Unggul)

UNTAR untuk INDONESIA

DATABASE DESIGN

PROGRAM STUDI TEKNIK INFORMATIKA/
SISTEM INFORMASI

UNIVERSITAS TARUMANAGARA

CONCEPTUAL DESIGN

Design Methodology

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

- **Database design methodology has 3 main phases:**
  - **Conceptual database design;**
  - **Logical database design;**
  - **Physical database design.**

Conceptual/Logical Database Design

- **Conceptual database design**
  - **Process of constructing a model of information used in an enterprise, independent of *all* physical considerations.**

- **Logical database design**
  - **Process of constructing a model of information 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

- **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.**

Methodology Overview - Conceptual Database Design

- **Step 1 Build conceptual data model for each user view**
  - **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 and primary key attributes**
  - **Step 1.6 Consider use of enhanced modeling concepts (optional step)**
  - **Step 1.7 Check model for redundancy**
  - **Step 1.8 Validate local conceptual model against user transactions**
  - **Step 1.9 Review local conceptual data model with user**

Methodology Overview - Logical Database Design for Relational Model

- **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 Check 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**

Methodology Overview - Physical Database Design for Relational Databases

- **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 enterprise 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**

Methodology Overview - Physical Database Design for Relational Databases

- **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 Model

- Objective: To build a conceptual data model of the data requirements of the enterprise.
- The first step in conceptual database design is to build one (or more) conceptual data models of the data requirements of the enterprise. A conceptual data model comprises:
  - entity types;
  - relationship types;
  - attributes and attribute domains;
  - primary keys and alternate keys;
  - integrity constraints.

Step 1 Build Conceptual Data Model

- **Step 1.1 Identify entity types**
  - **To identify the main entity types that are required by the view.**
- **Step 1.2 Identify relationship types**
  - **To identify the important relationships that exist between the entity types that have been identified.**
- **Step 1.3 Identify and associate attributes with entity or relationship types**
  - **To identify and associate attributes with the appropriate entity or relationship types and document the details of each attribute.**

Step 1 Build Conceptual Data Model

- **Step 1.4 Determine attribute domains**
  - **To determine domains for the attributes in the local conceptual model and document the details of each domain**

- **Step 1.5 Determine candidate and primary 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.**

- **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.**
- **Step 1.8 Validate local conceptual model against user transactions**
  - **To ensure that the local conceptual model supports the transactions required by the view.**
- **Step1.9   Review local conceptual data model with user**
  - **To review the local conceptual data model with the user to ensure that the model is a ‘true’ representation of the user’s view of the enterprise.**

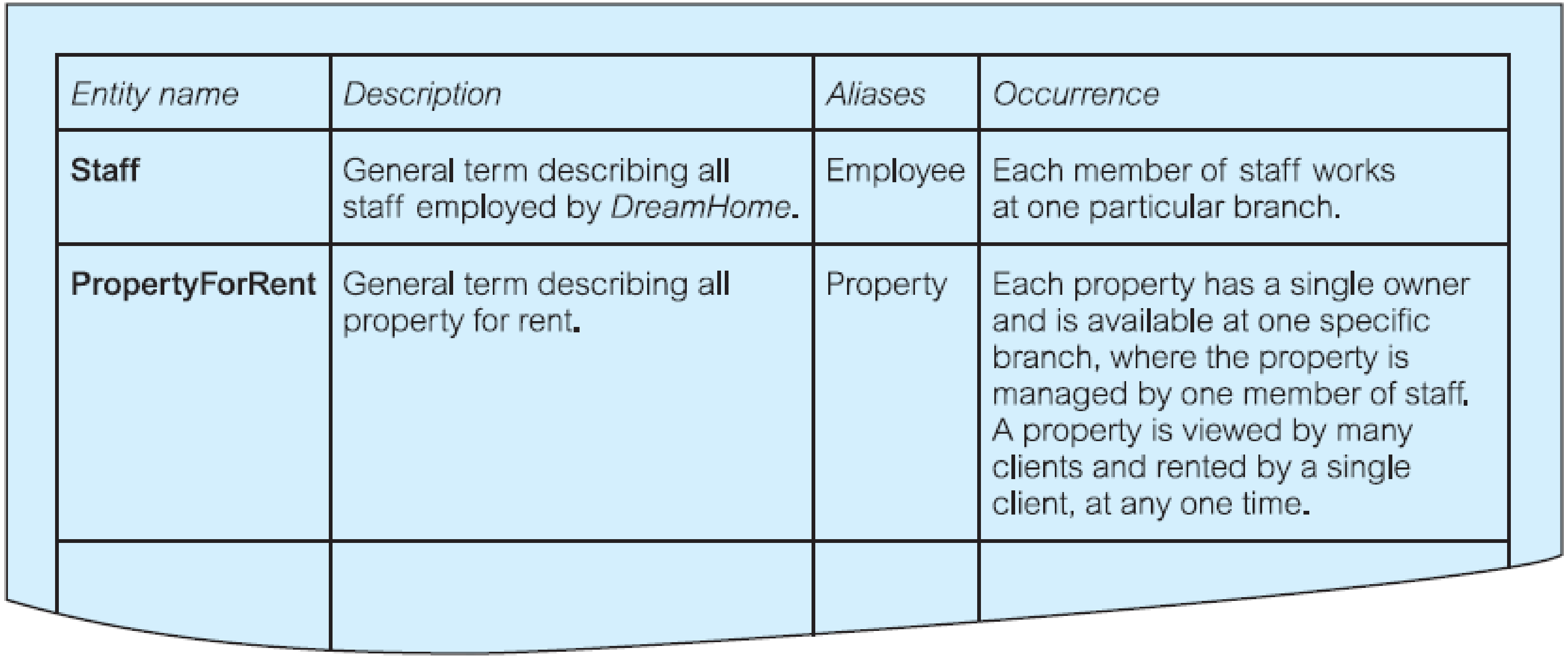
Use examples from the Staff user views of the DreamHome case Appendix A

- **For the Staff user views of DreamHome we identify the following entities:**
  - **Staff**
  - **PropertyForRent**
  - **PrivateOwner**
  - **BusinessOwner**
  - **Client**
  - **Preference**
  - **Lease**

Step 1.1 Identify entity types

Extract from Data Dictionary for Staff View of *DreamHome* Showing Description of Entities

Document entity types

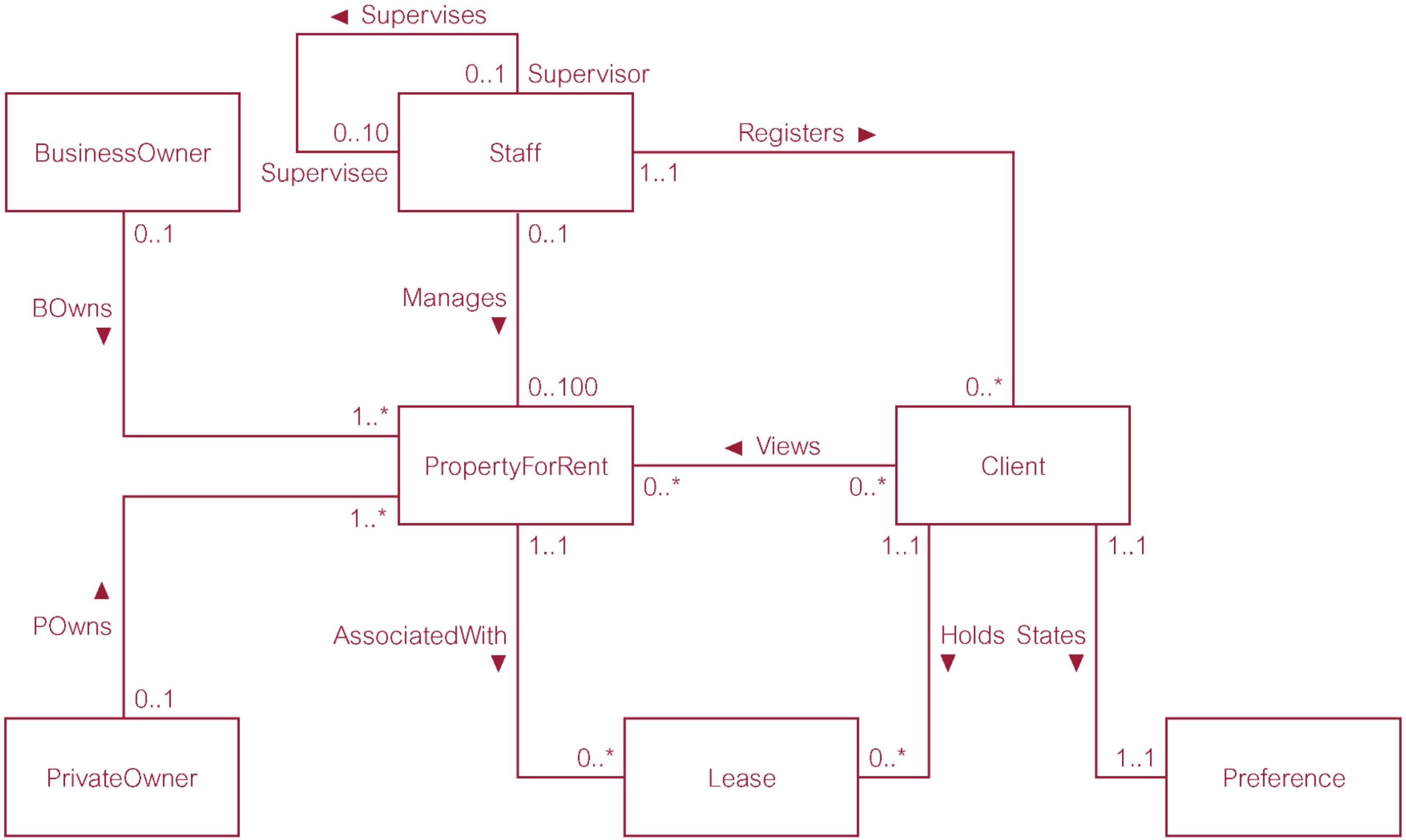


Entity name	*Description*	*Aliases*	*Occurrence*
Staff	General term describing all staff employed by *DreamHome*.	Employee	Each member of staff works at one particular branch.
PropertyForRent	General term describing all property for rent.	Property	Each property has a single owner and is available at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

Step 1.2 Identify relationship types

- Use Entity–Relationship (ER) diagrams
- Determine the multiplicity constraints of relationship types
- Check for fan and chasm traps

Step 1.2 Identify relationship types

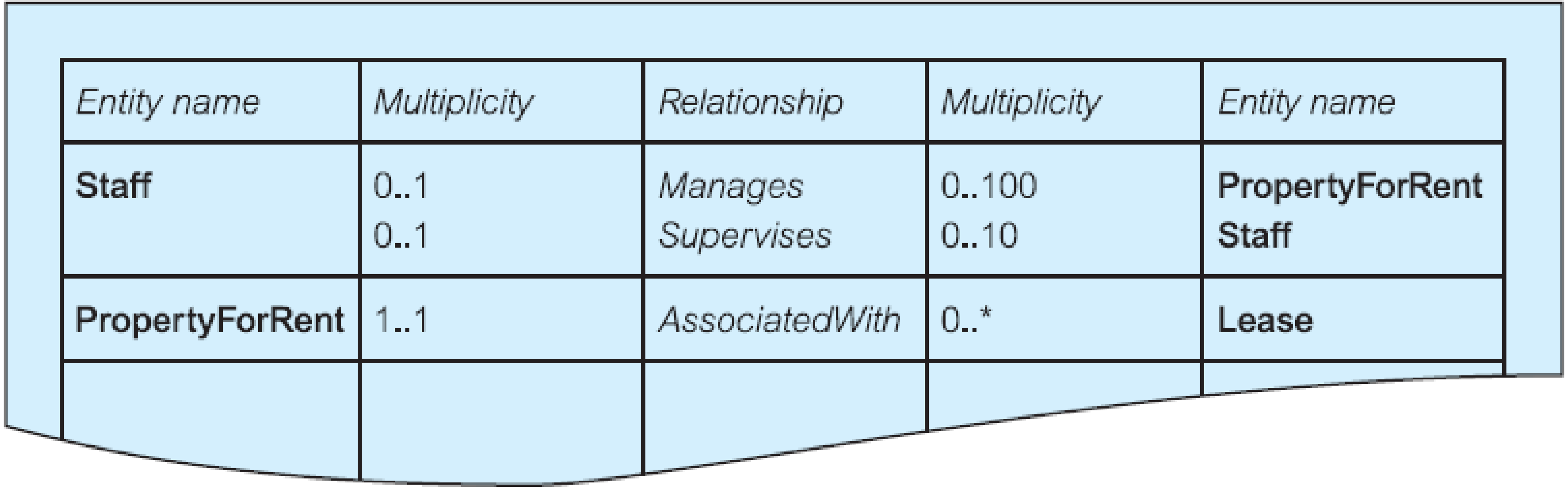


First-cut ER diagram for Staff View of *DreamHome*

Step 1.2 Identify relationship types

Extract from Data Dictionary for Staff View of *DreamHome* Showing Description of Relationships

Document relationship types



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**

Step 1.3 Identify and associate attributes with entity or relationship types

The next step in the methodology is to identify the types of facts about the entities and relationships that we have chosen to be represented in the database.

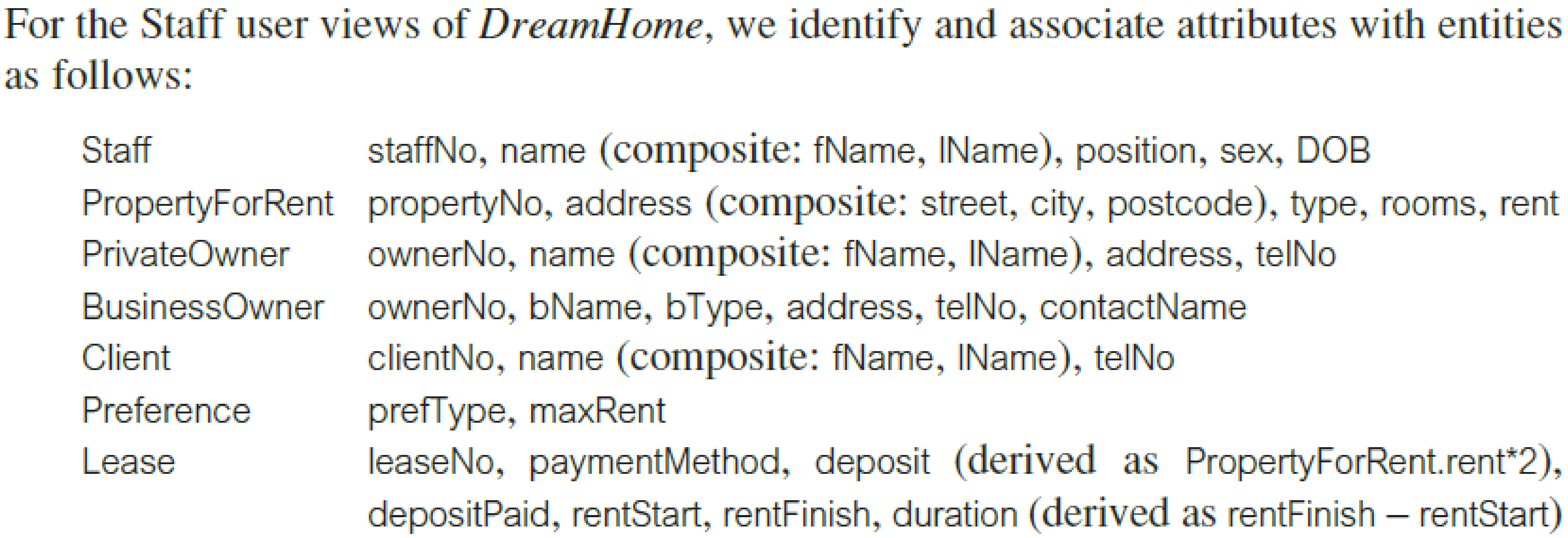
- Simple/composite attributes
- Single/multi-valued attributes
- Derived attributes

Step 1.3 Identify and associate attributes with entity or relationship types

- Potential problems
  - For example, we may have identified the entities Staff and PropertyForRent with the following attributes:
    - Staff staffNo, name, position, sex, DOB
    - PropertyForRent propertyNo, street, city, postcode, type, rooms, rent, managerName
  - The presence of the managerName attribute in PropertyForRent is intended to represent the relationship Staff Manages PropertyForRent. In this case, the managerName attribute should be omitted from PropertyForRent and the relationship Manages should be added to the model.

Step 1.3 Identify and associate attributes with entity or relationship types

- DreamHome attributes for entities



For the Staff user views of *DreamHome*, we identify and associate attributes with entities as follows:

Staff	staffNo, name (composite: fName, lName), position, sex, DOB
PropertyForRent	propertyNo, address (composite: street, city, postcode), type, rooms, rent
PrivateOwner	ownerNo, name (composite: fName, lName), address, telNo
BusinessOwner	ownerNo, bName, bType, address, telNo, contactName
Client	clientNo, name (composite: fName, lName), telNo
Preference	prefType, maxRent
Lease	leaseNo, paymentMethod, deposit (derived as PropertyForRent.rent*2), depositPaid, rentStart, rentFinish, duration (derived as rentFinish – rentStart)

Step 1.3 Identify and associate attributes with entity or relationship types

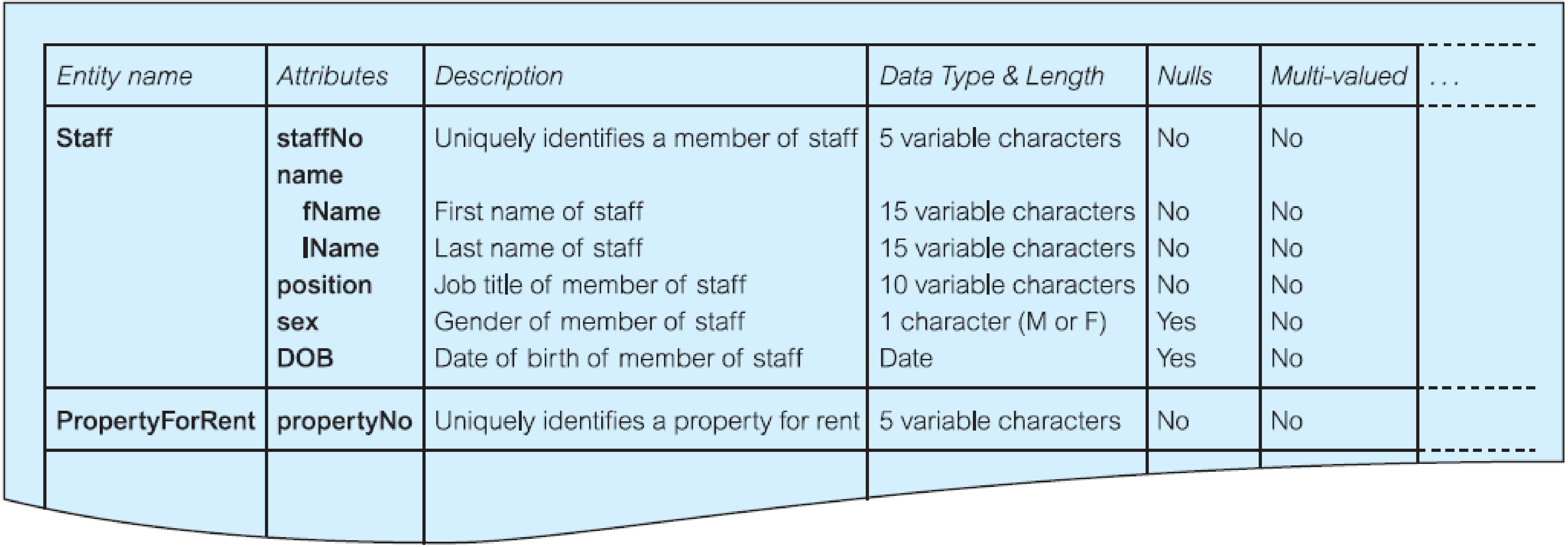
- DreamHome attributes for relationships

  - For the Staff user views of DreamHome, we identify and associate attributes with relationships as follows:
    - Views (viewDate, comment)

Step 1.3 Identify and associate attributes with entity or relationship types

Document attributes

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



Entity name	*Attributes*	*Description*	*Data Type & Length*	*Nulls*	*Multi-valued*	*. . .*
Staff	**staffNo**	Uniquely 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**	Uniquely identifies a property for rent	5 variable characters	No	No	

Step 1.4 Determine attribute domains

A fully developed data model specifies the domains for each attribute and includes:

- allowable set of values for the attribute;
- n sizes and formats of the attribute.

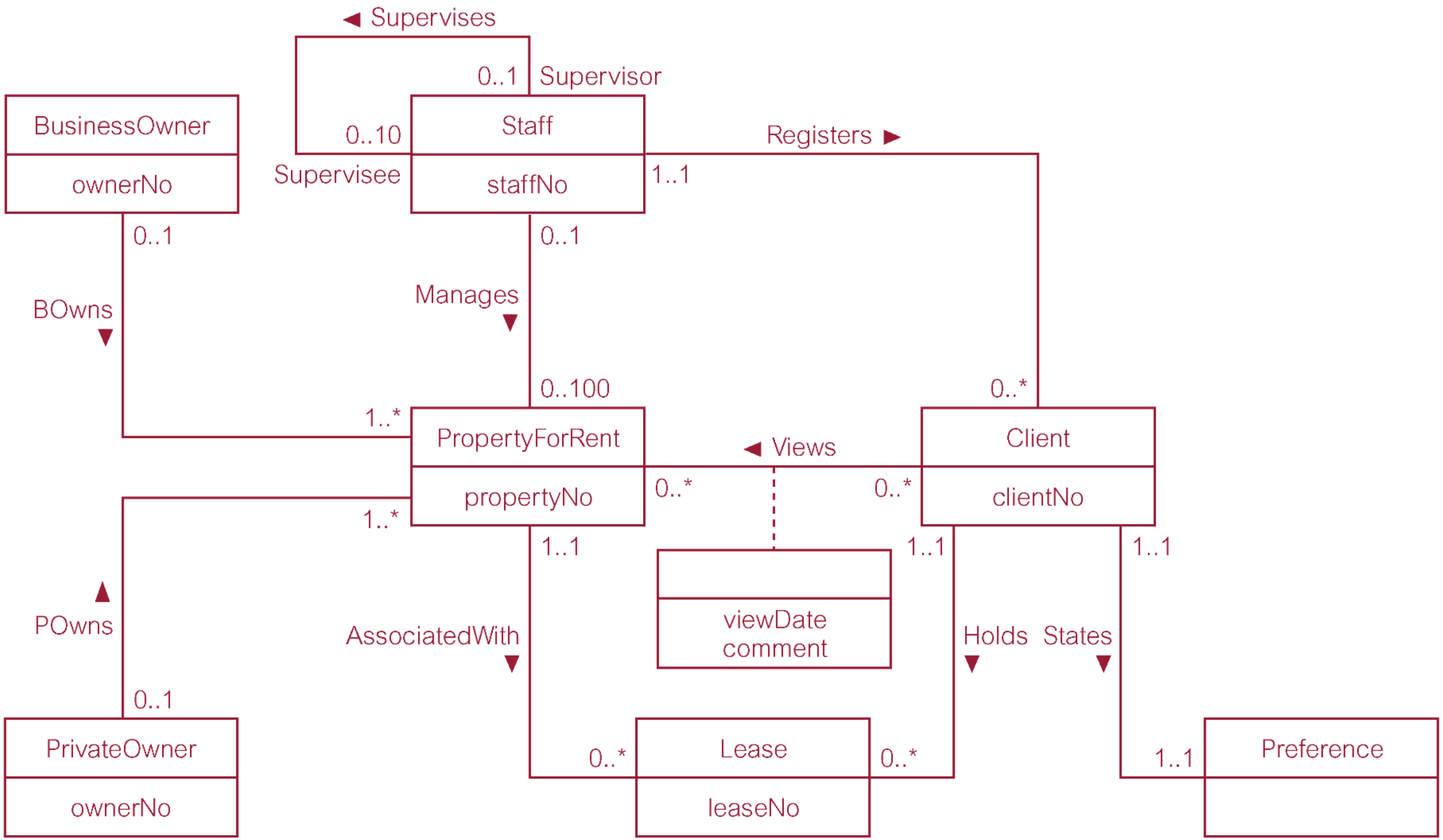
Step 1.5 Determine candidate, primary, and alternate key attributes

- When choosing a primary key from among the candidate keys, use the following guidelines to help make the selection:
  - the candidate key with the minimal set of attributes;
  - the candidate key that is least likely to have its values changed;
  - the candidate key with fewest characters (for those with textual attribute(s));
  - the candidate key with smallest maximum value (for those with numerical attribute(s));
  - the candidate key that is easiest to use from the users' point of view.

Step 1.5 Determine candidate, primary, and alternate key attributes

- When choosing a primary key from among the candidate keys, use the following guidelines to help make the selection:
  - the candidate key with the minimal set of attributes;
  - the candidate key that is least likely to have its values changed;
  - the candidate key with fewest characters (for those with textual attribute(s));
  - the candidate key with smallest maximum value (for those with numerical attribute(s));
  - the candidate key that is easiest to use from the users' point of view.

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



Step 1.6 Consider use of enhanced modeling concepts (optional step)

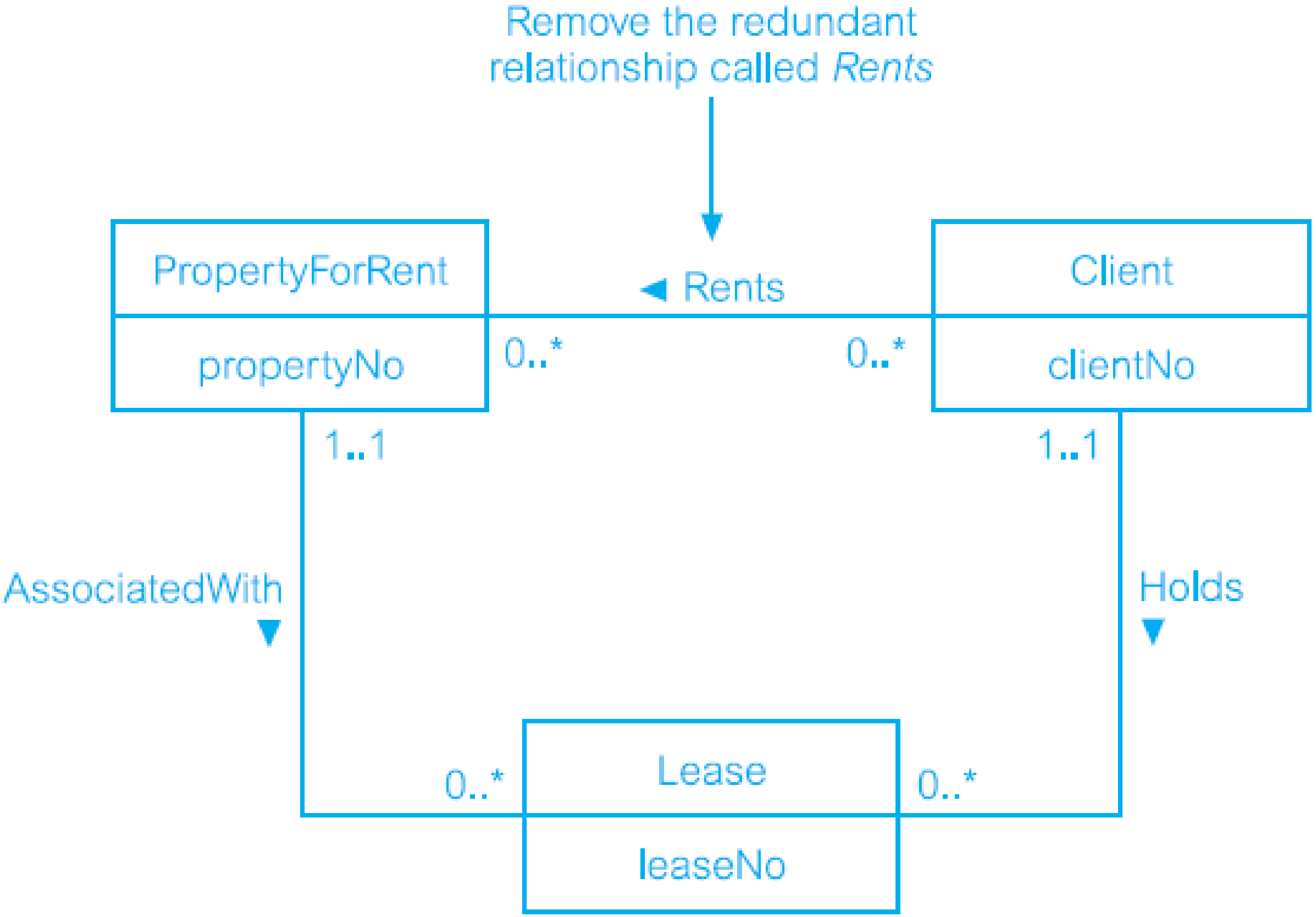
Step 1.7 Check model for redundancy

- re-examine one-to-one (1:1) relationships;
- remove redundant relationships;
- consider time dimension

Step 1.7 Check model for redundancy

- EXAMPLE: remove redundant relationships;

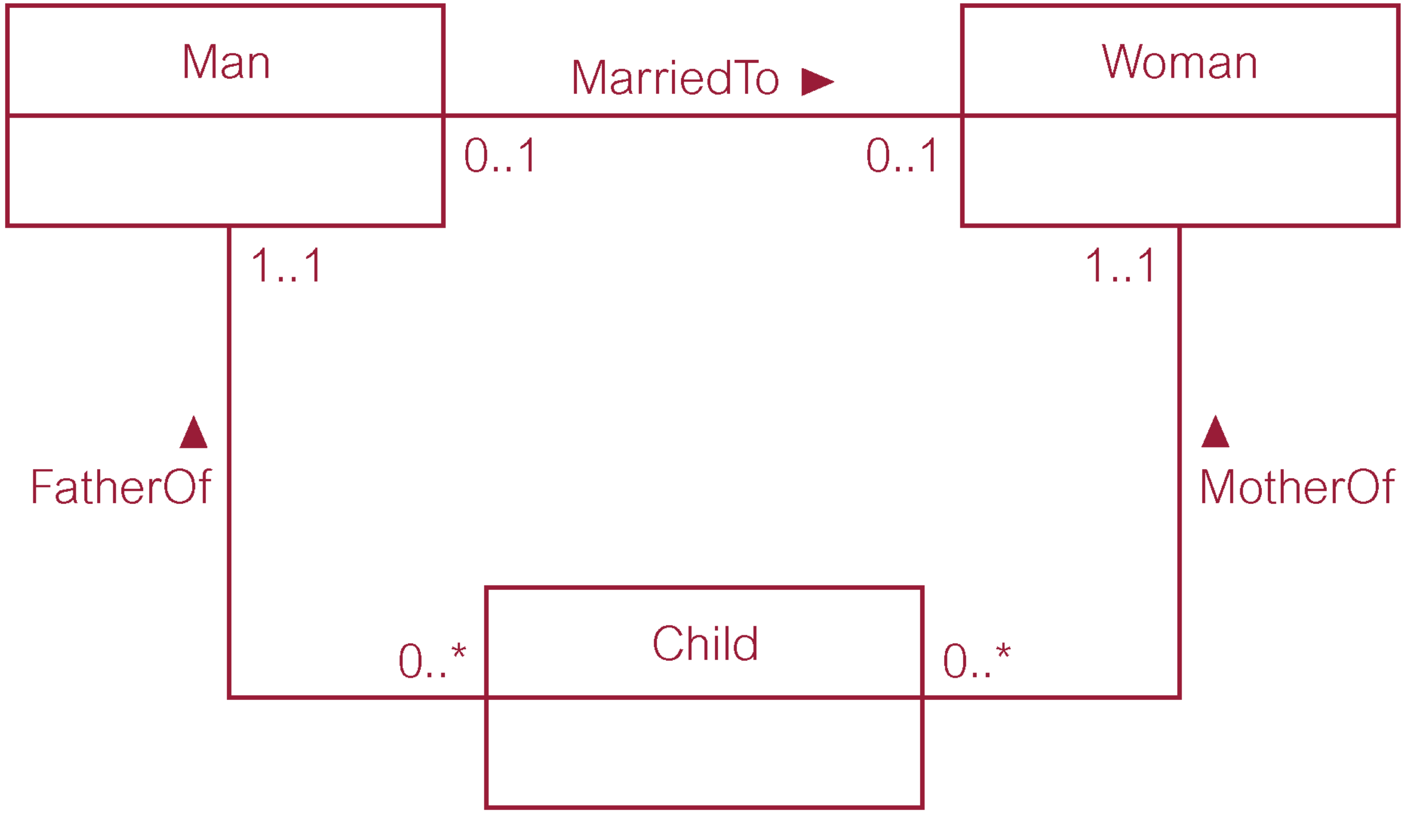
Remove the redundant relationship called Rents.



Step 1.7 Check model for redundancy

- EXAMPLE: consider time dimension;

Example of a non-redundant relationship FatherOf.



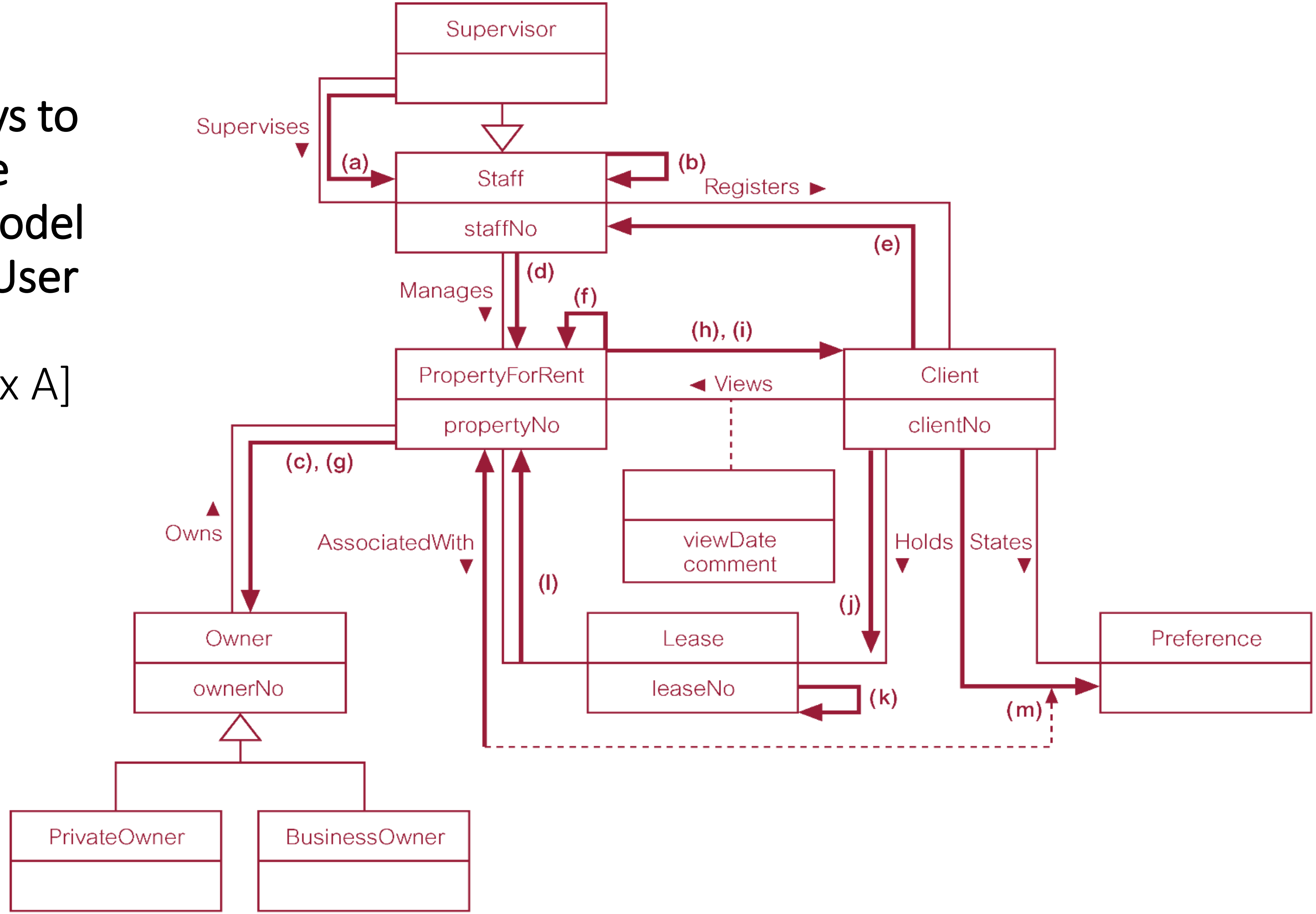
Step 1.8 Validate conceptual model against user transactions

- We examine two possible approaches to ensuring that the conceptual data model supports the required transactions:

  - describing the transactions;
  - using transaction pathways

Using Pathways to Check that the Conceptual Model Supports the User Transactions

[lihat appendix A]



Exercises

The EasyDrive School of Motoring case study - Appendix B.2.

15.18 Create a conceptual data model for the case study. State any assumptions necessary to support your design. Check that the conceptual data model supports the required transactions.

Thank You

Reference: Database Systems A Practical Approach to Design, Implementation, and Management Fourth Edition.

Thomas M. Connolly and Carolyn E. Begg