Chapter 16

Methodology
Conceptual Databases Design

Chapter 16 - Objectives

- 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.

Chapter 16 - Objectives

- 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.

Chapter 16 - Objectives

- 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 modelling 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 modelling 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 for the 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 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 for relational database

- **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**

Overview Database Design Methodology

- 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

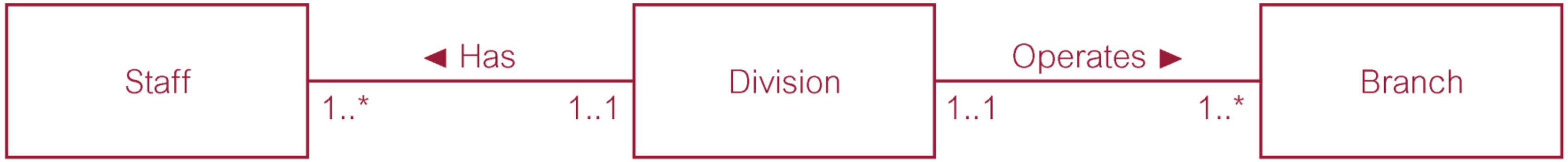
Step 1 Build Conceptual Data

- 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.

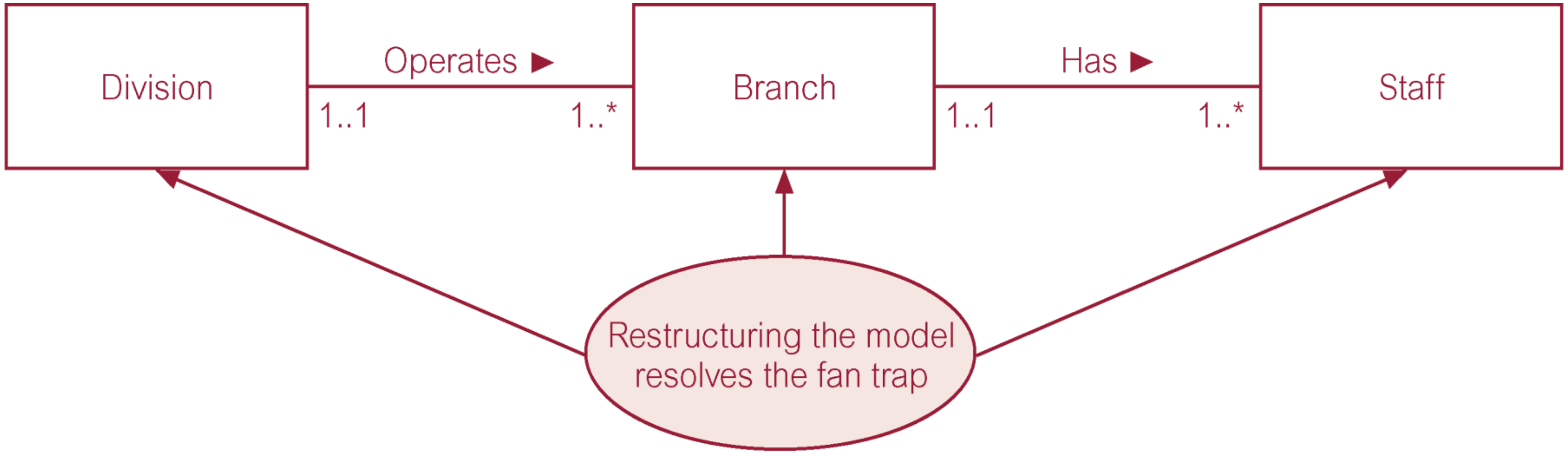
- Step 1.1 Identify entity types
  - To identify the required entity types.
  - Identify by Nouns in the requirements.

- Step 1.2 Identify relationship types
  - To identify the important relationships that exist between the entity types.
  - Identified by finding appropriate verbs in the requirements.
- Check for:
  - Fan trap – pathway between entities is ambiguous, often happen in a many-to-many relationship
  - Chasm trap - the model suggest a pathway, but non exists

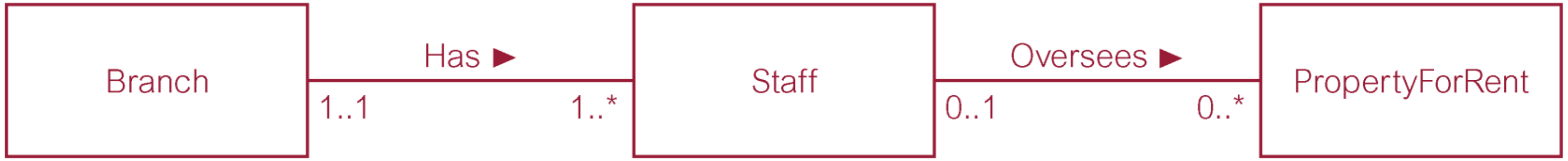
Fan trap



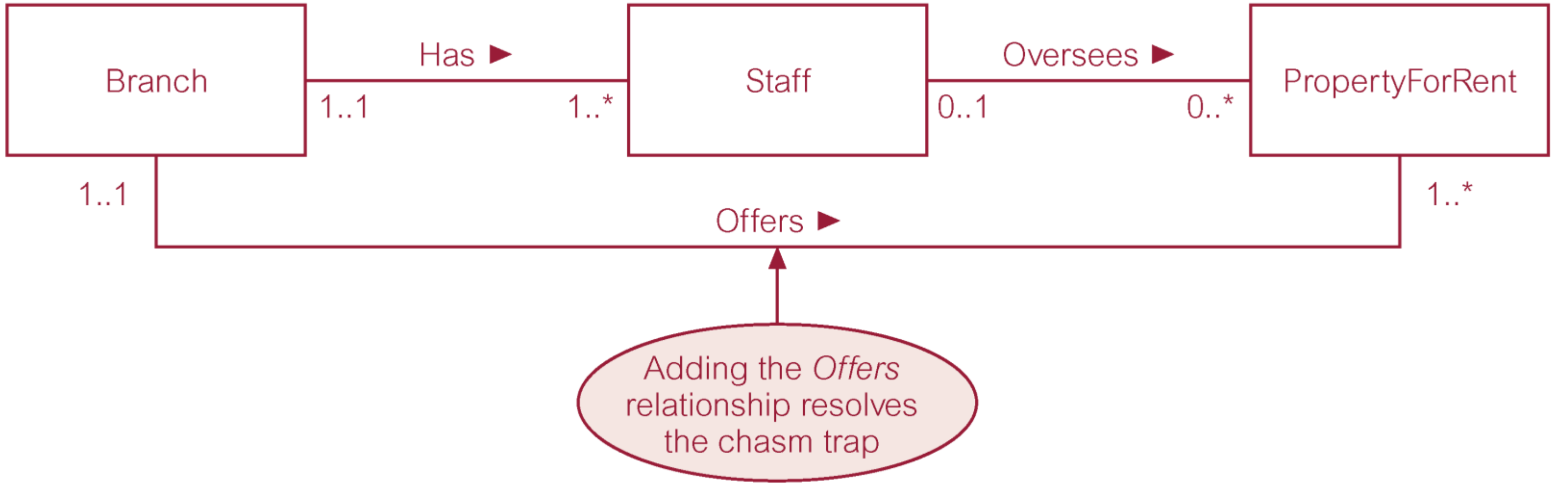
- Can we answere the question:
- At which branch does the staff member SM10?



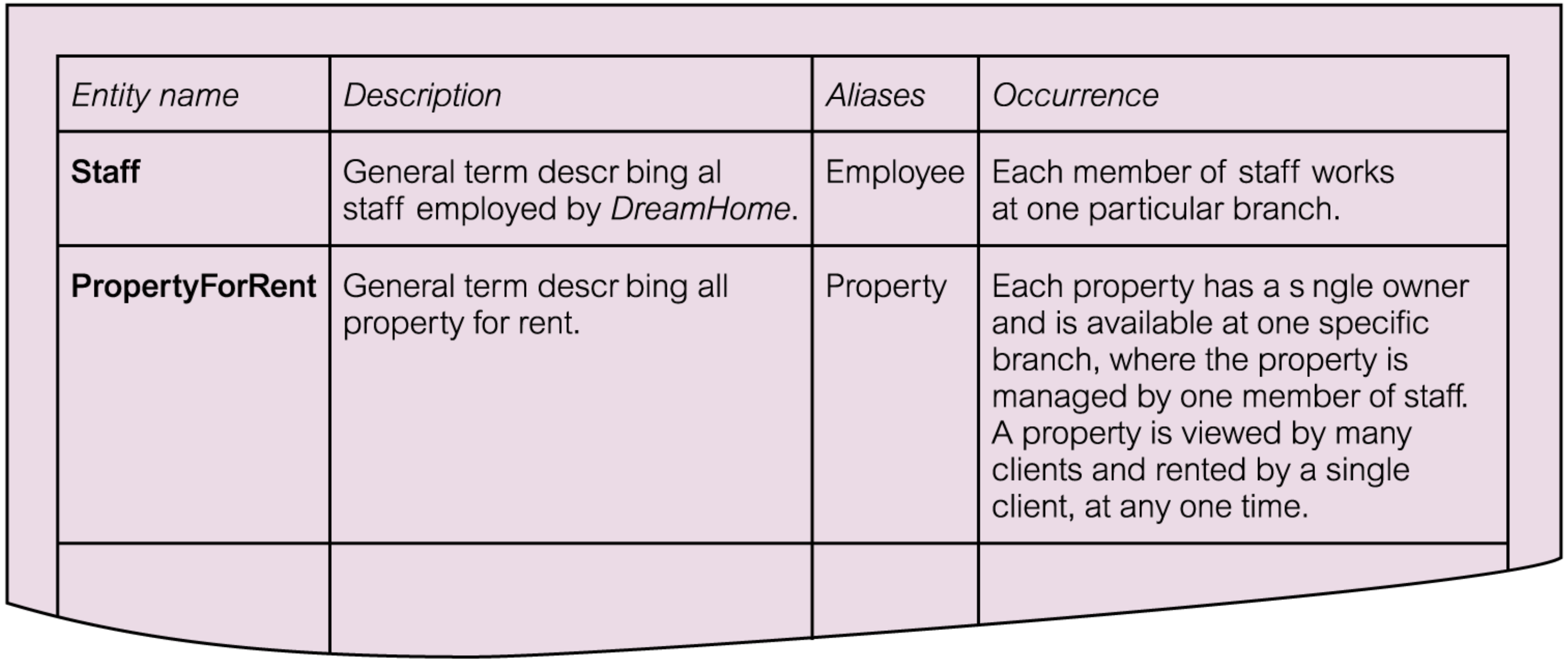
Chasm trap



- A property exists in the area of a given branch, but since the property is not allocated to a staff yet, it is not available.
- At which branch is property P1 allocated?

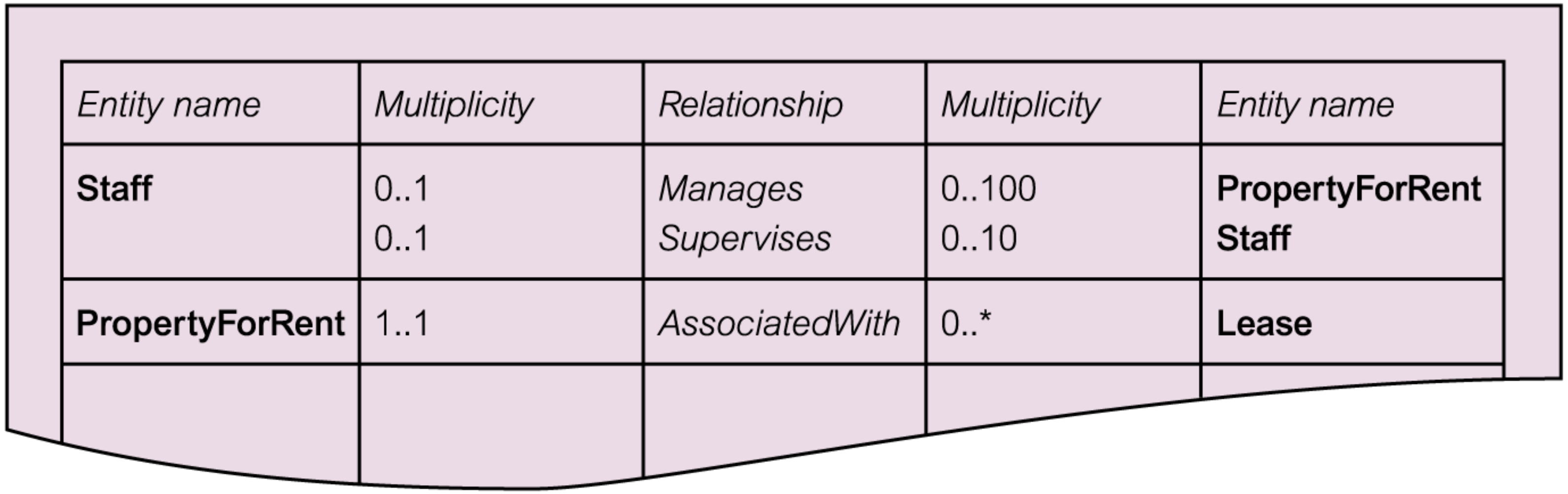


Extract from data dictionary for Staff user views of *DreamHome* showing description of entities



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.

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**

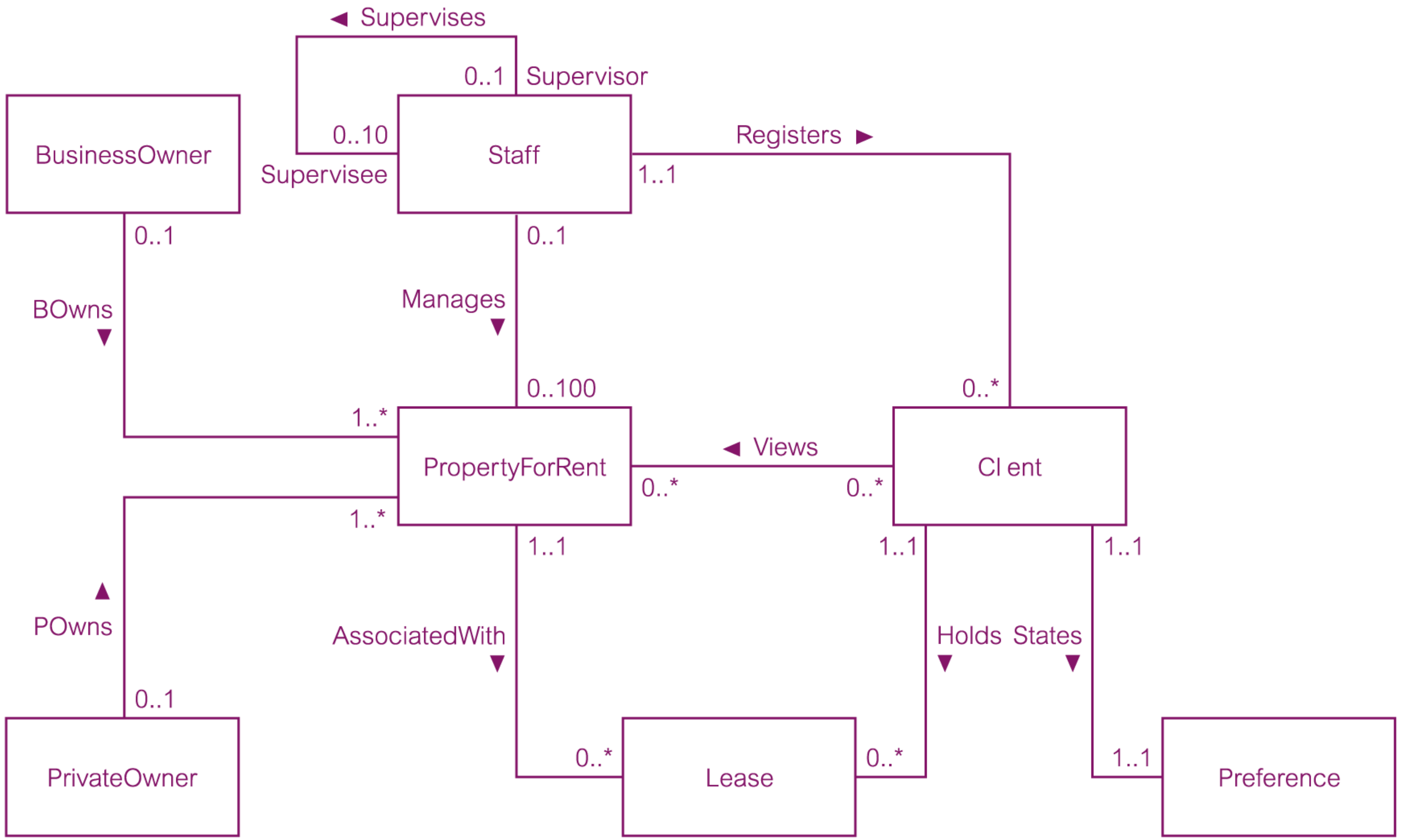
Step 1 Build Conceptual Data

- Entities
  - Staff
  - PropertyForRent
  - PrivateOwner
  - BusinessOwner
  - Client
  - Preference
  - Lease

- Relationship
  - Staff Manages PropertyForRent
  - PrivateOwner Owns PropertyForRent
  - PropertyForRent AssociatedWith Lease

Make the EA model using class diagram

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



Step 1 Build Conceptual Data - attributes

- 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.
  - The attributes can be identified where the noun or noun phrase is a property, quality, identifier, or characteristic of one of these entities or relationships
  - ask “What information are we required to hold on x or y?”
  - The answer to this question should be described in the specification.

Attributes

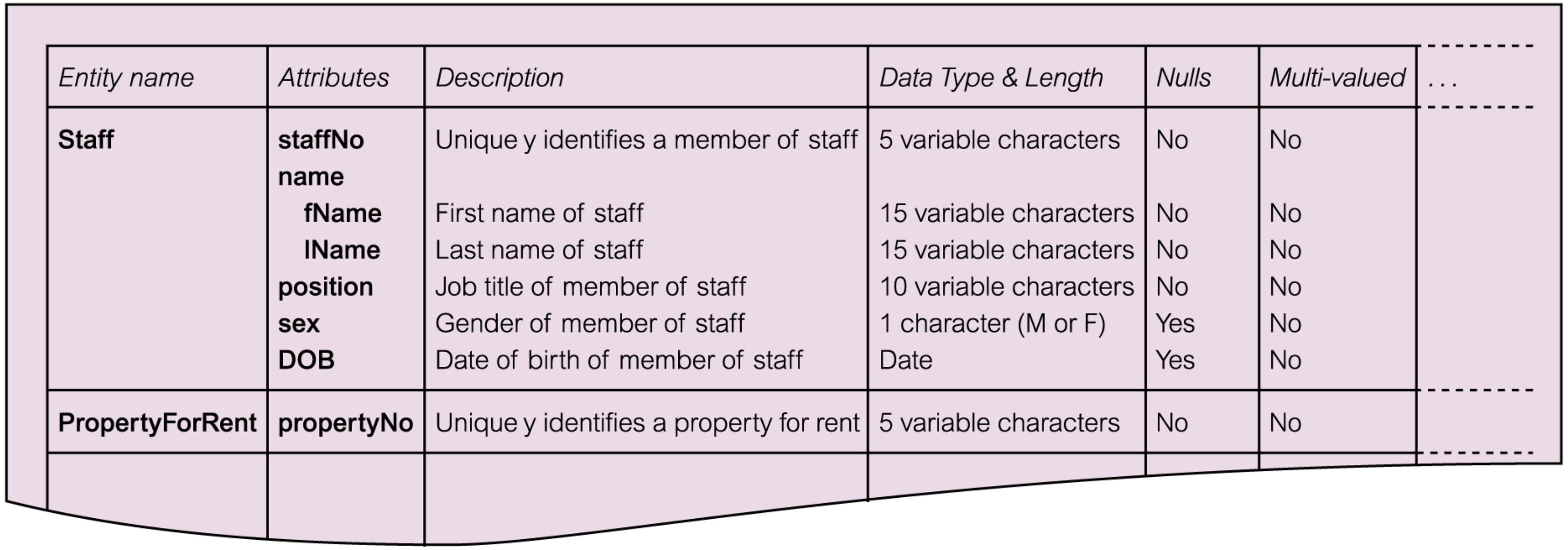
- Simple/Complex attributes
  - Simple
    - Age
  - Complex attributes
    - Address
- Single/Multi-valued attributes
  - Most are single
  - Some, like telephone number, may be multiple
- Derived attributes
  - Age
  - Number of properties
  - Rental deposit (f.ex twice the monthly rate)

Dream home attributes

- 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, eMail
- Preference prefType, maxRent
- Lease leaseNo, paymentMethod, deposit (derived as PropertyForRent.rent*2), depositPaid, rentStart, rentFinish, duration (derived as rentFinish – rentStart)

- Attribute for relationship
  - View viewDate, comment

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	

Step 1 Build Conceptual Data

- Step 1.4 Determine attribute domains
  - To determine domains for the attributes in the data model and document the details of each domain.
  - A domain is a pool of values from which one or more attributes draw their values
  - SSN (social security number - Norway)
    - 6 digit birthdate
    - 3 digit for individual number, the third shall be even values for female, uneven value for male. Given sequentially within the birth date.
    - 2 digit control number
  - Sex – M, F, O

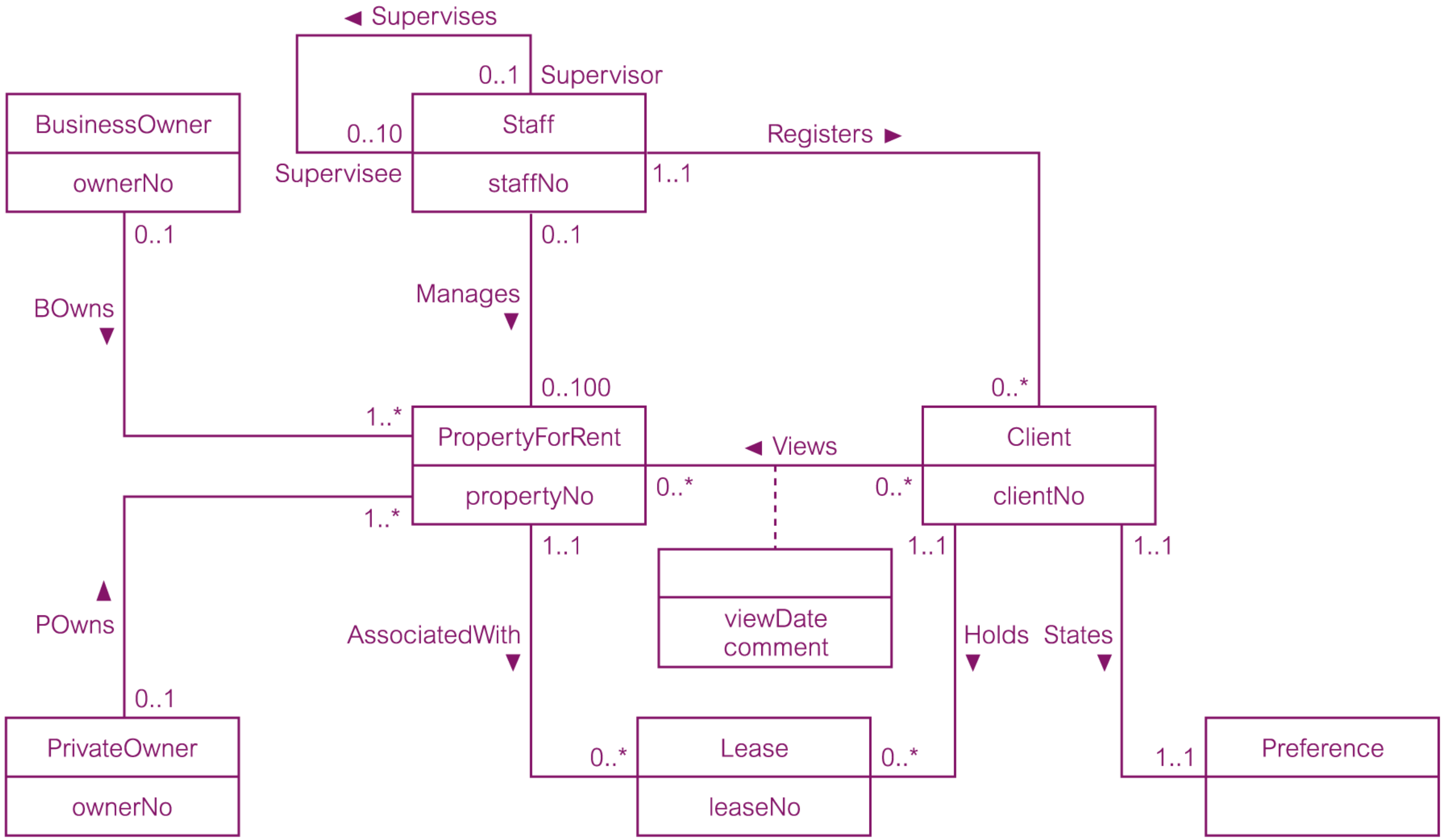
Step 1 Build Conceptual Data – primary key

- **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.
  - To identify the candidate key(s) for each entity type and, if there is more than one candidate key, to choose one to be the primary key and the others as alternate keys.
  - Bad example of primary key:
    - Peoples name – “sorry, can’t hire you, we already have an employee with that name”

Step 1 Build Conceptual Data – primary key

- 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
- Strong vs weak entity
  - If you can assign an entity a primary key - the entity is referred to as being strong.
  - If you are unable to assign an entity a primary key – the entity is referred to as being weak.
  - Come back to this in step 2.1
- Let's update our model with primary

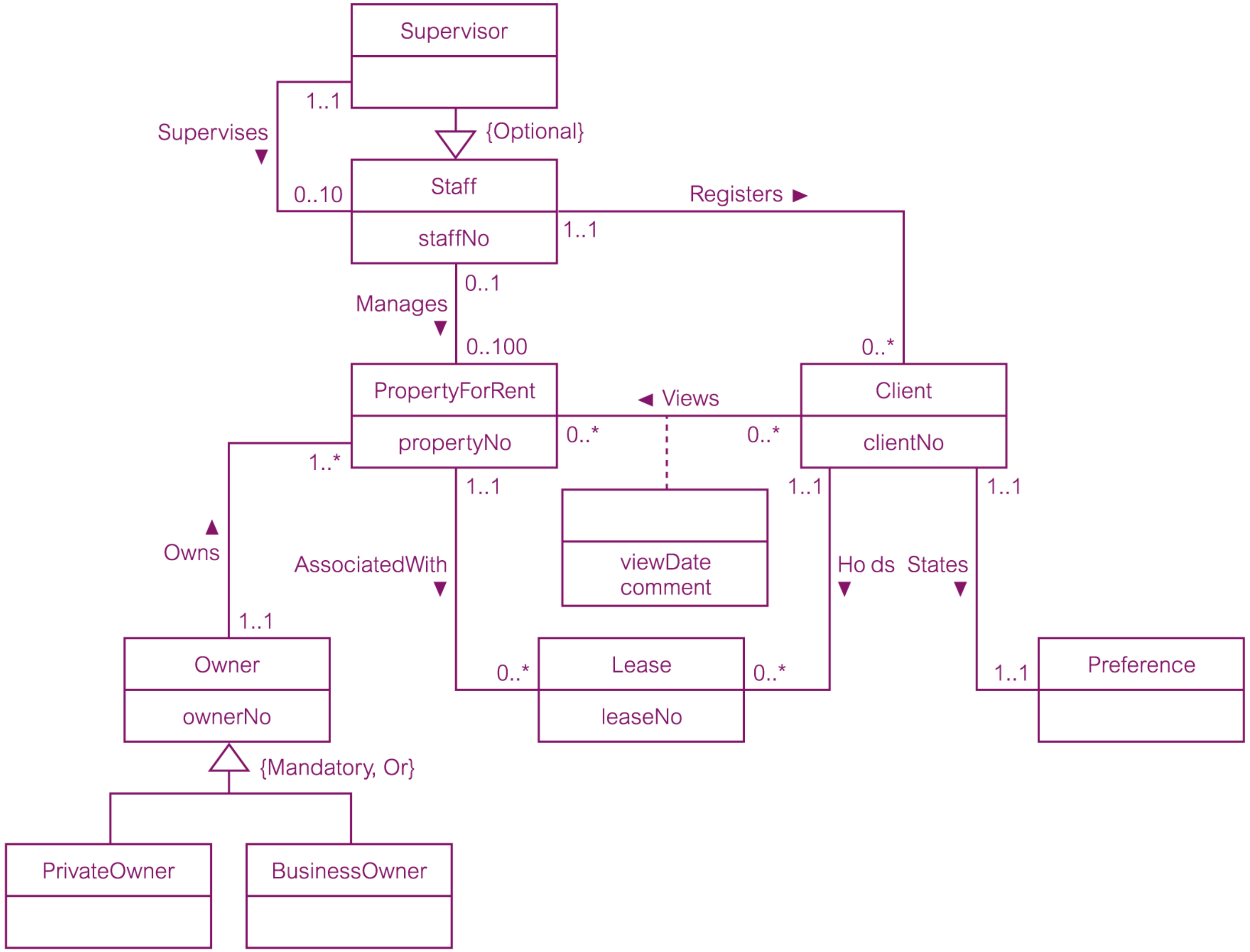
ER diagram for Staff user views of *DreamHome* with primary keys added



Step 1 Build Conceptual Data

- 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.
- Are there any such enhancement in our model?
- Do we have some entities that are similar – that share a lot of data?
  - Business owner and private owner
  - Staff and Supervisor
- Let’s update our model

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

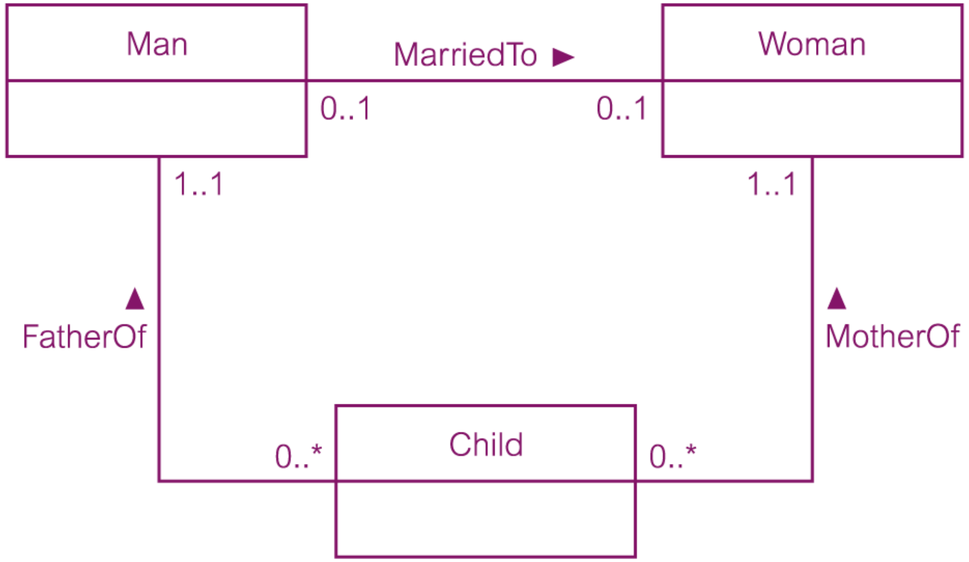


Step 1 Build Conceptual Data Model

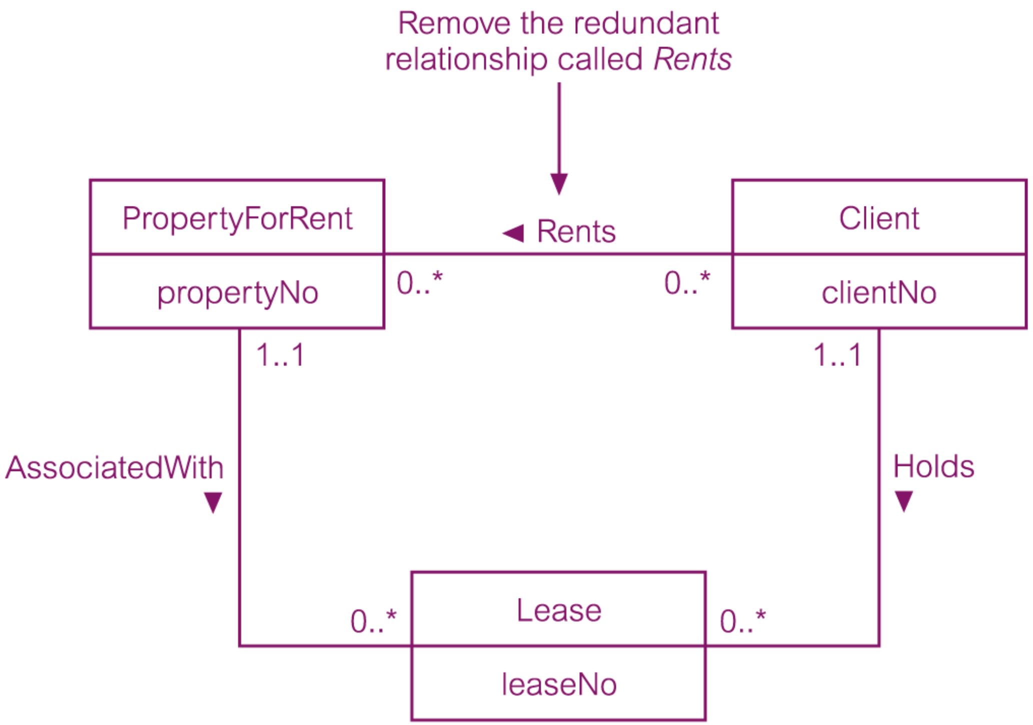
- Step 1.7 Check model for redundancy
  - To check for the presence of any redundancy in the model and to remove any that does exist.
- Three activities in this step are:
  - (1) re-examine one-to-one (1:1) relationships
  - (2) remove redundant relationships
  - (3) consider time dimension.
    - E.g., when things change over time. This is usually very troublesome for many of the students.

Consider time dimension

- Some would try to simplify and remve the fatherOf relationship.
- What if the mother and father are no longer married?
- What if the Woman has children from before?... Or the man has?



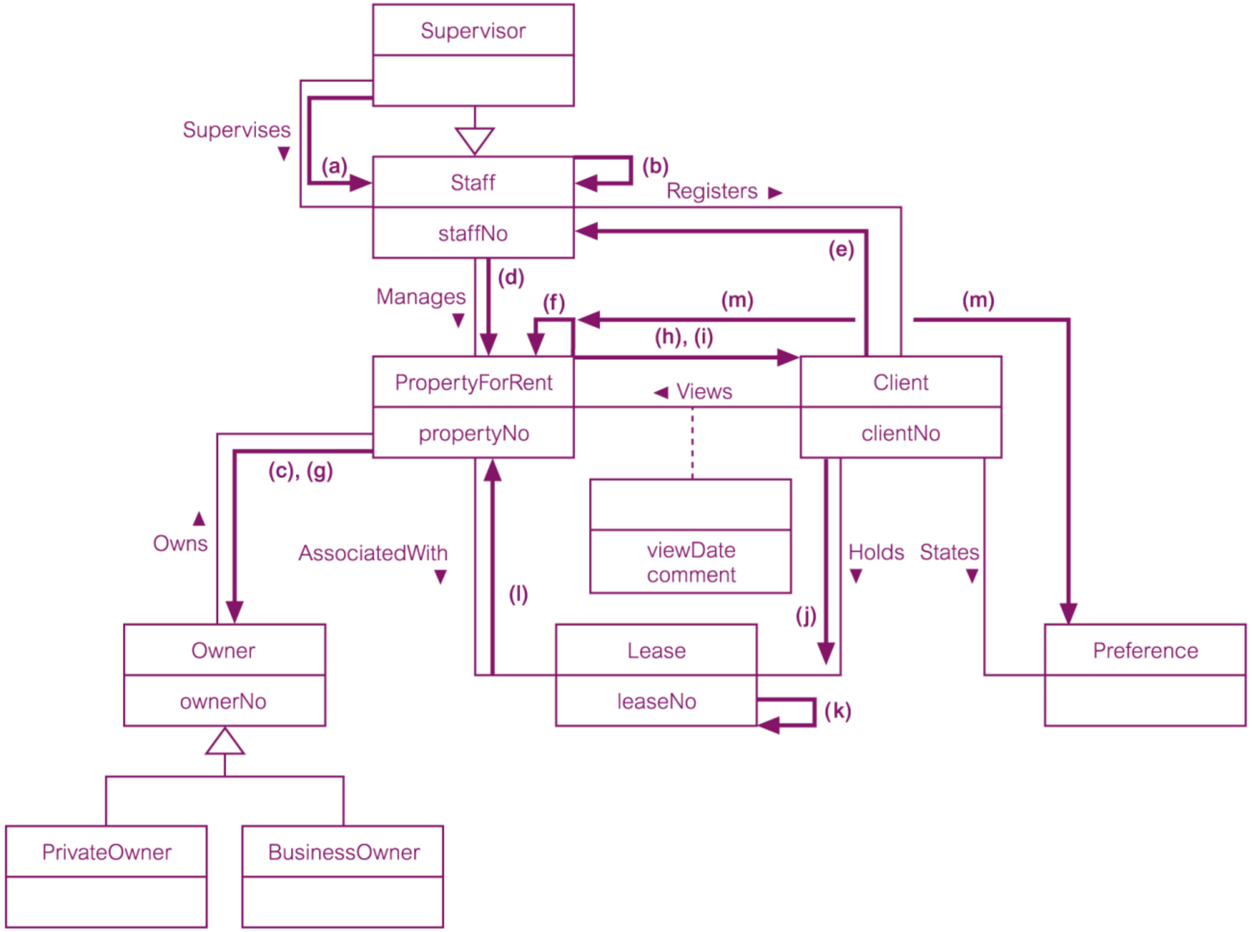
Example of removing a redundant relationship called *Rents*


Remove the redundant relationship called *Rents*

Step 1 Build Conceptual Data Model

- Step 1.8 Validate conceptual model against user transactions
  - To ensure that the conceptual model supports the required transactions.
- Two possible approaches to ensuring that the conceptual data model supports the required transactions:
  - (1) describing the transactions
  - (2) using transaction pathways.
- Transaction (d): List the details of properties managed by a named member of staff at the branch

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



Step 1 Build Conceptual Data Model

- 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.