CS 422 - DBMS

Lesson 6 - Chapter 12

Entity-Relationship Modeling

The three in one structure of the Unified Field

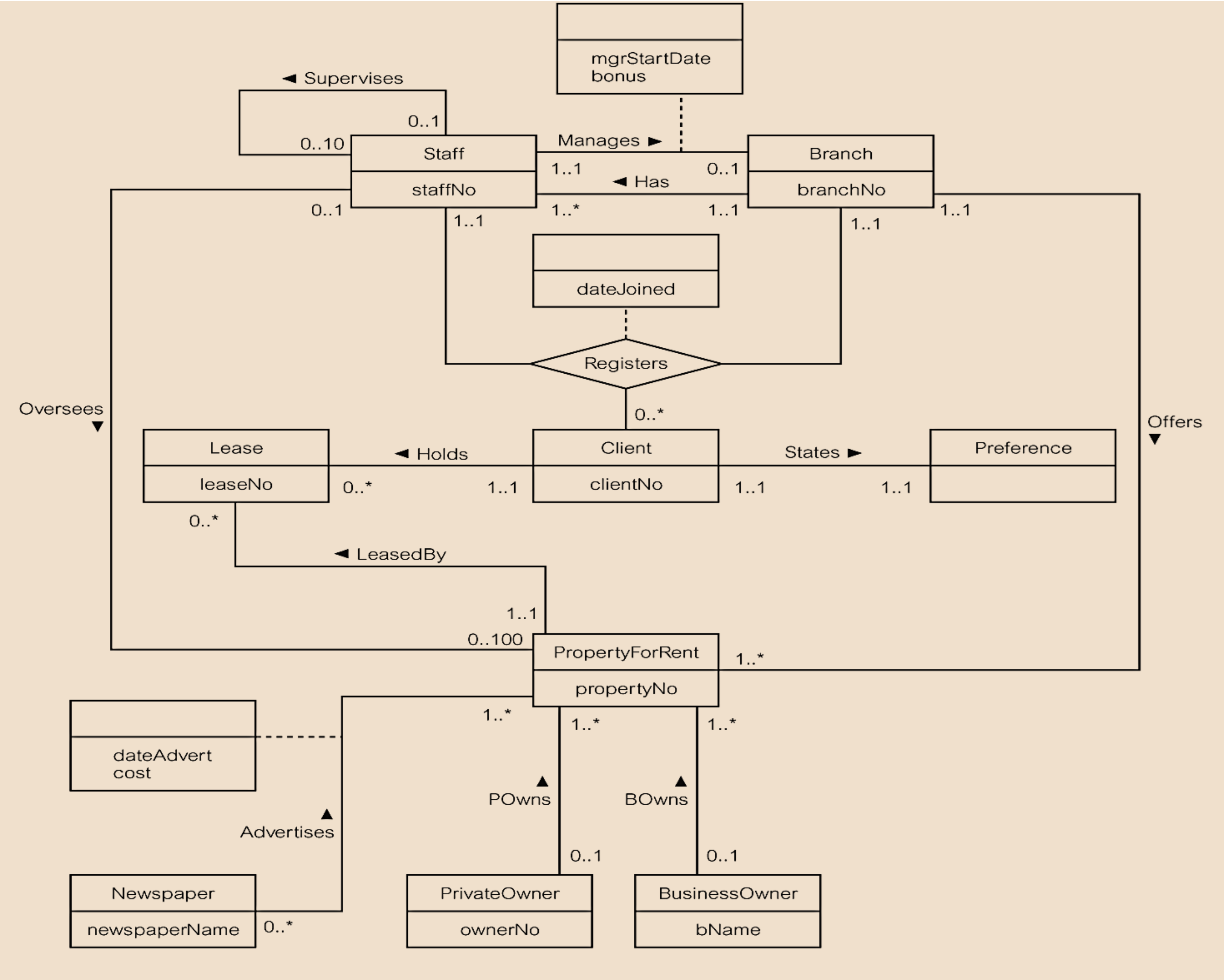
WHOLENESS OF THE LESSON

The entity-relationship data model is based on a perception of the real world that consists of a set of basic objects called entities, and of relationships among these entities. Entities and relationships can have attributes. Science & Technology of Consciousness: Knowledge is perceived differently in different states of consciousness. In unity consciousness, one perceives all objects in terms of the Self.

Chapter 12 - Objectives

- How to use Entity–Relationship (ER) modeling in database design.
- How to build an ER model from a requirements specification.
- Basic concepts associated with ER model.
- Diagrammatic technique for displaying ER model using Unified Modelling Language (UML).
- How to identify and resolve problems with ER models called connection traps.

ERD of Branch user views of DreamHome



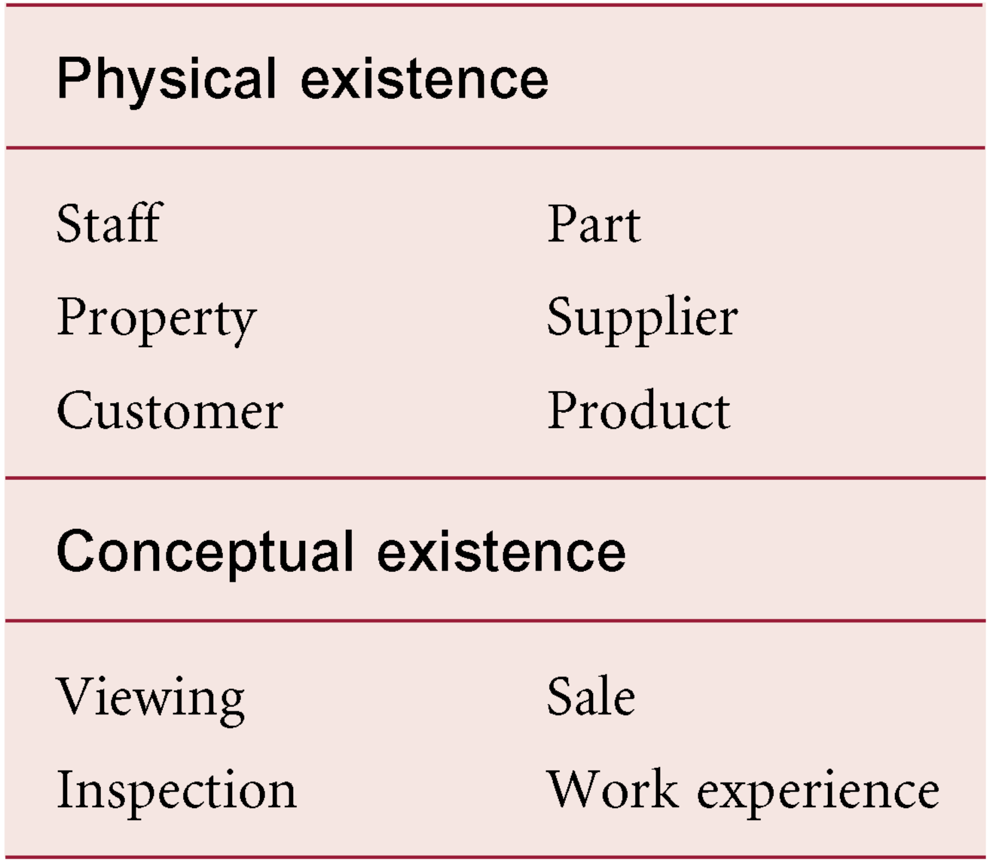
Concepts of the ER Model

- **Entity types**
- **Relationship types**
- **Attributes**

Entity Type

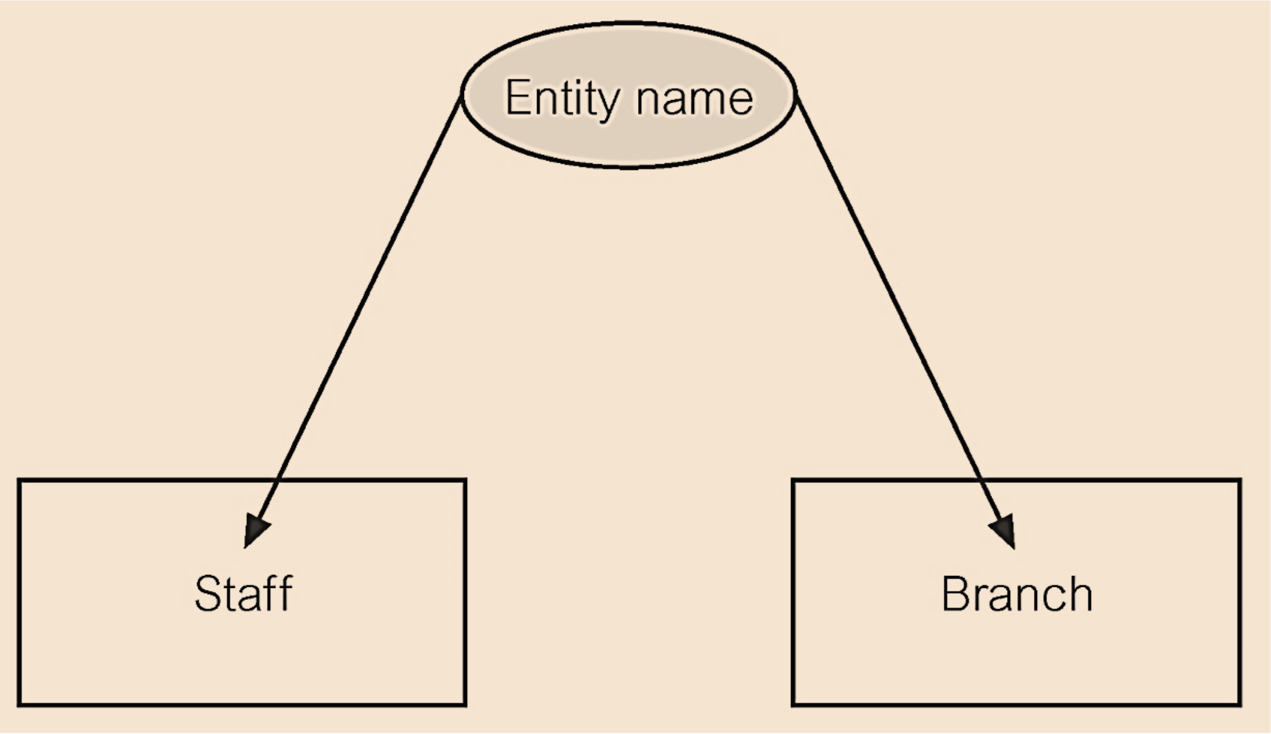
- **Entity type**
  - Group of objects with same properties, identified by enterprise as having an independent existence.
  - Represents a person, place or a thing that you want to track in a database.
- **Entity occurrence (entity instance)**
  - Uniquely identifiable object of an entity type.
  - This will be each record or row in a table.

Examples of Entity Types



Physical existence	
Staff	Part
Property	Supplier
Customer	Product
Conceptual existence	
Viewing	Sale
Inspection	Work experience

ER Diagram of Staff And Branch Entity Types

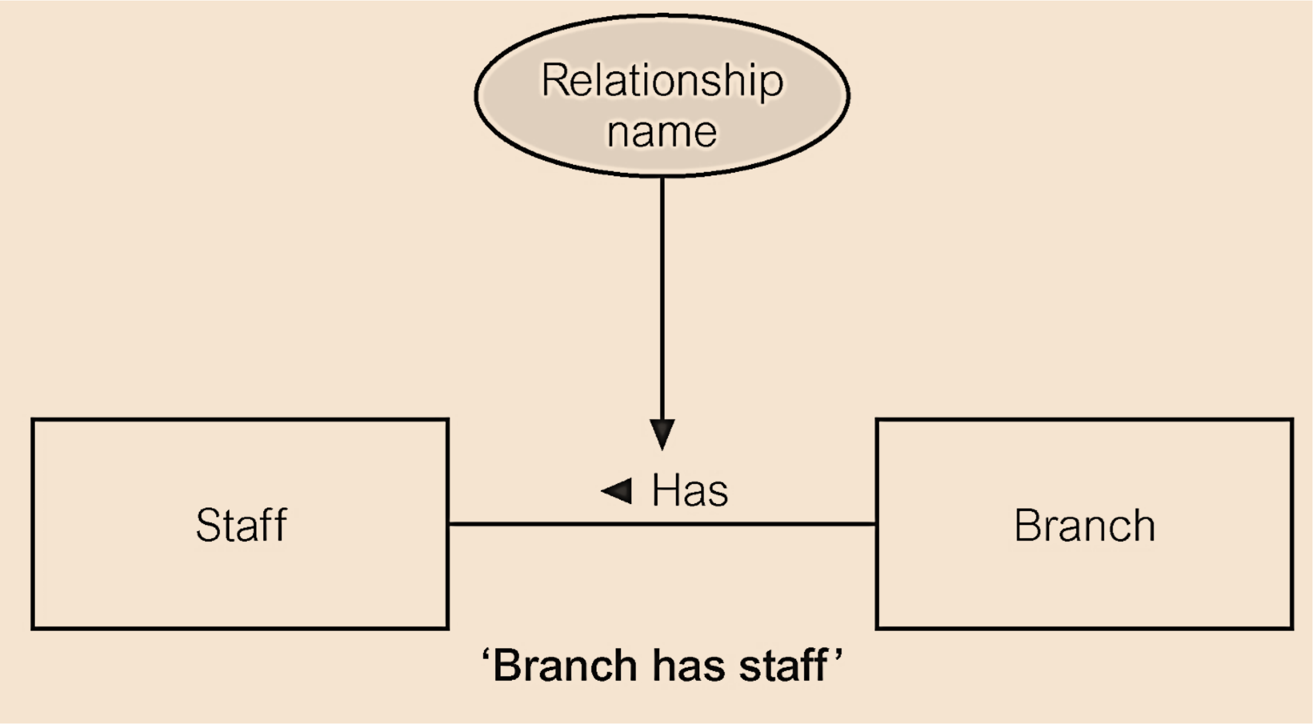


Relationship Types

- **Relationship type**
  - Set of meaningful associations among entity types.
  - Relationship describes how one or more entities interact with each other.
  - A verb is often used to describe the relationship.

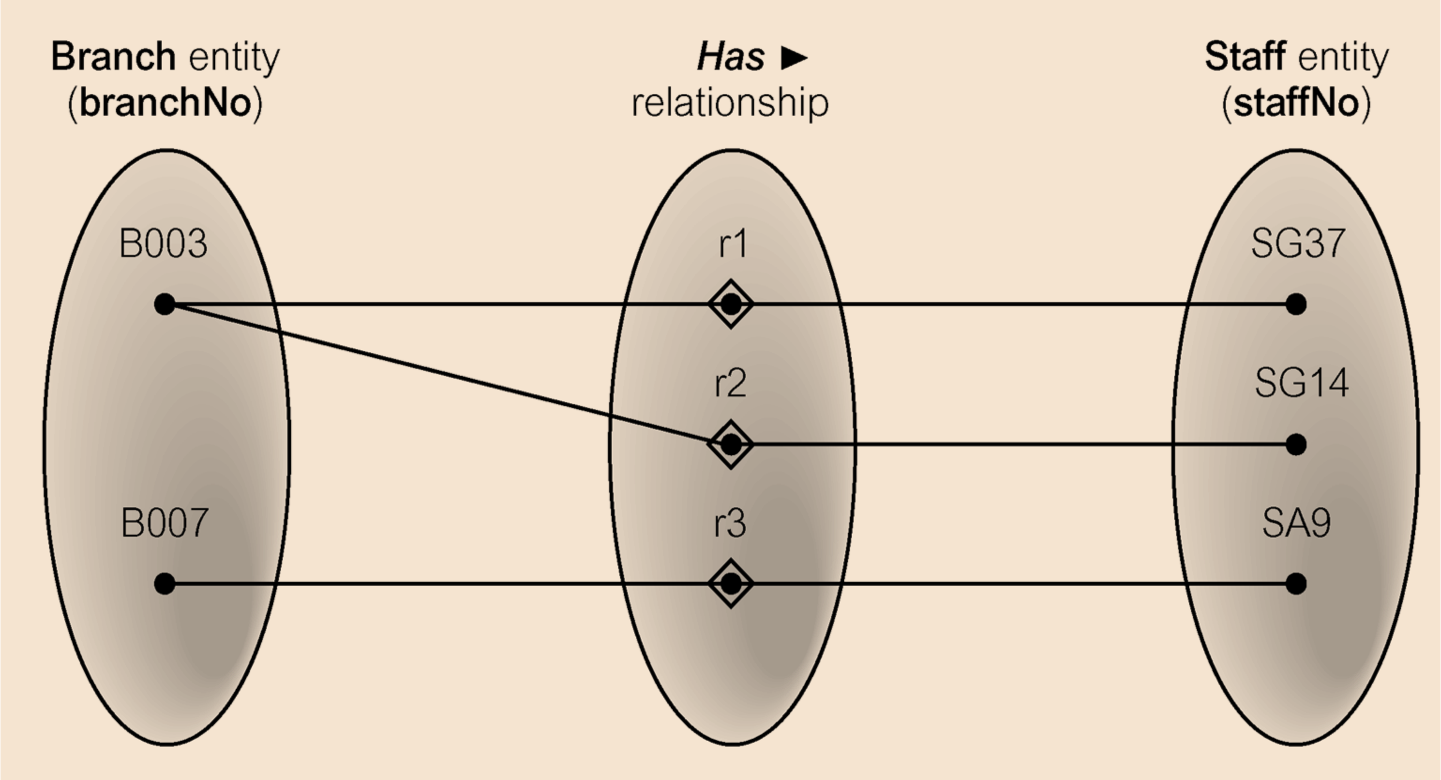
- **Relationship occurrence**
  - Uniquely identifiable association, which includes one occurrence from each participating entity type.

ER diagram of Branch Has Staff relationship



Semantic net of *Has* relationship type

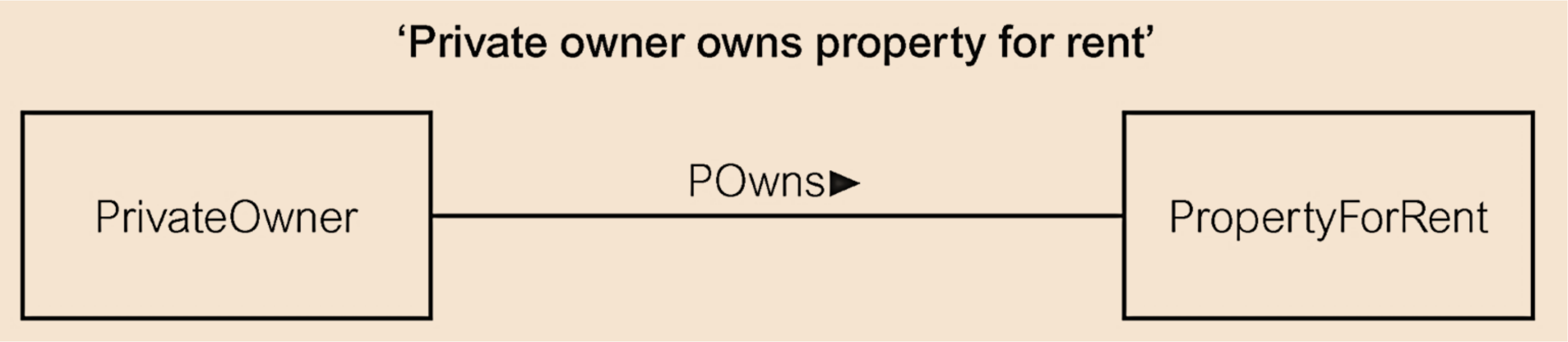
- Semantic net is an object level model.
- Entity type is Branch and entity occurrences are B003, B007. Same is the case with Staff.
- Relationship occurrences are r1, r2, r3

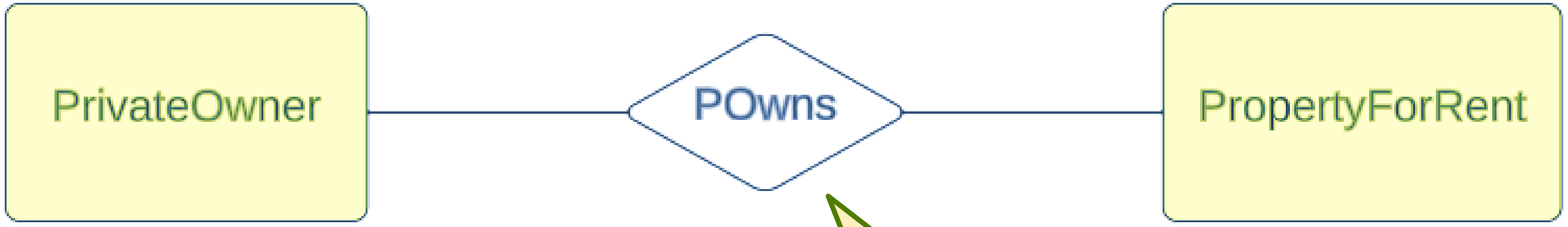


Degree of Relationship Type

- Degree of Relationship Type is the number of participating entity types in a relationship.
- A relationship of degree :
  - one is **unary**
  - two is **binary**
  - three is **ternary**
  - four is **quaternary**
- The term “complex relationship” is used to describe relationships with degrees higher than binary.

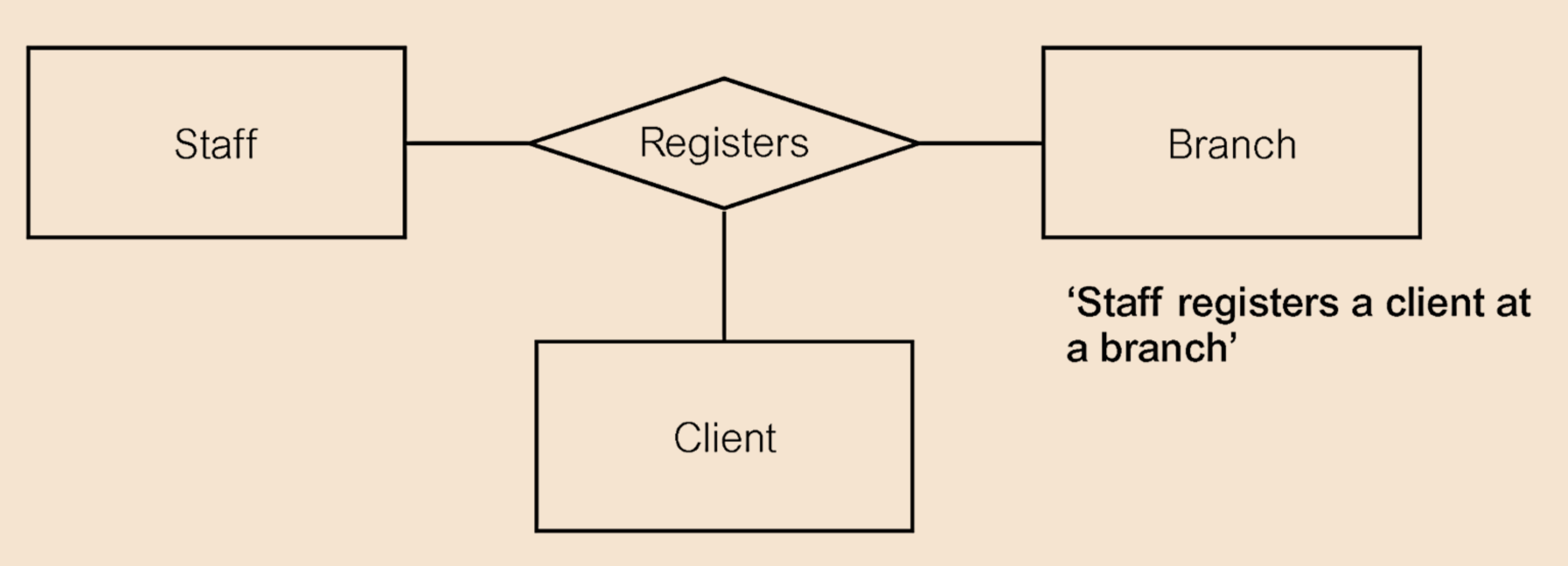
Binary relationship called *POwns*


‘Private owner owns property for rent’



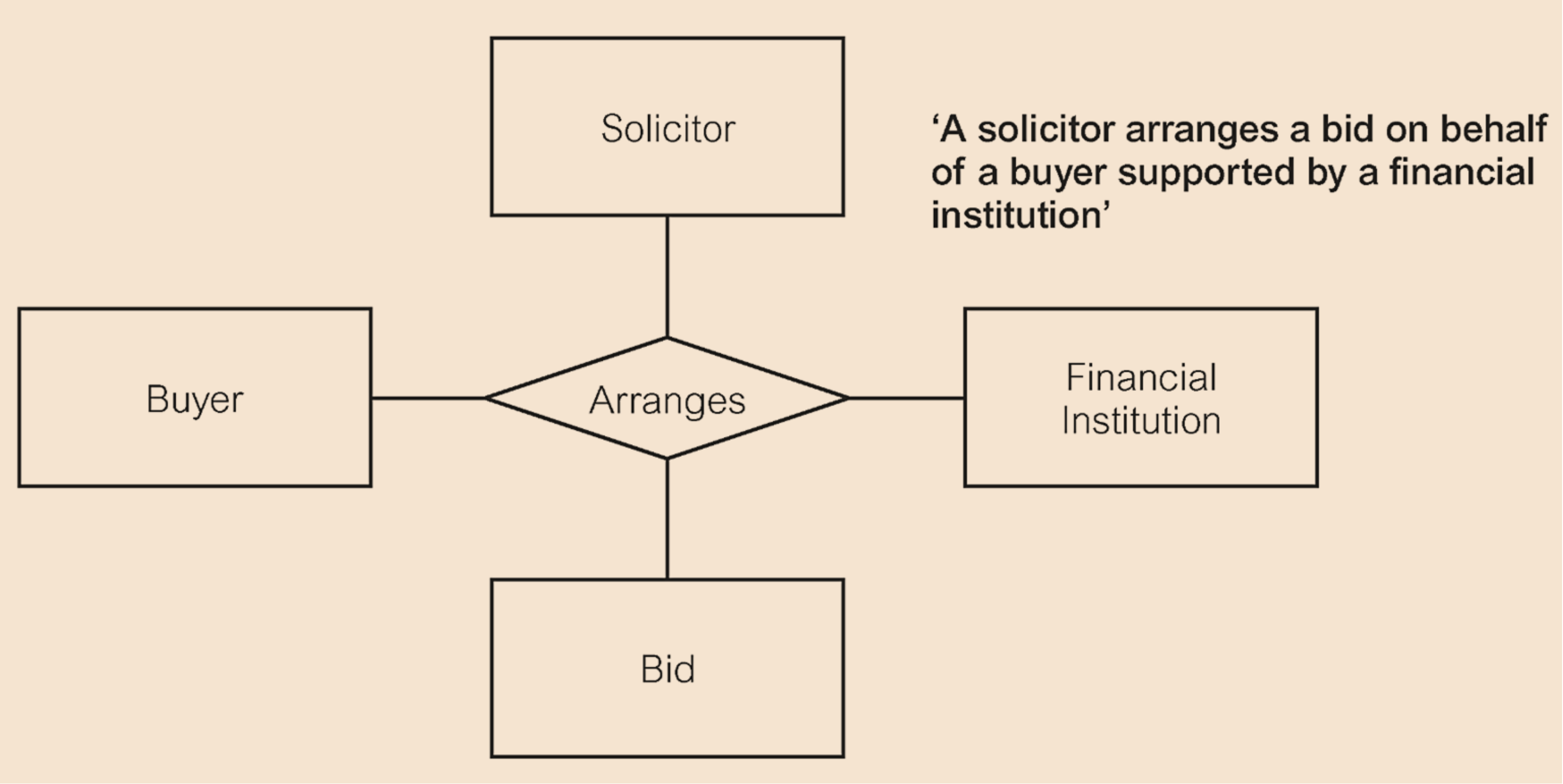
Different way of showing the relationship

Ternary relationship called *Registers*



'Staff registers a client at a branch'

Quaternary relationship called *Arranges*



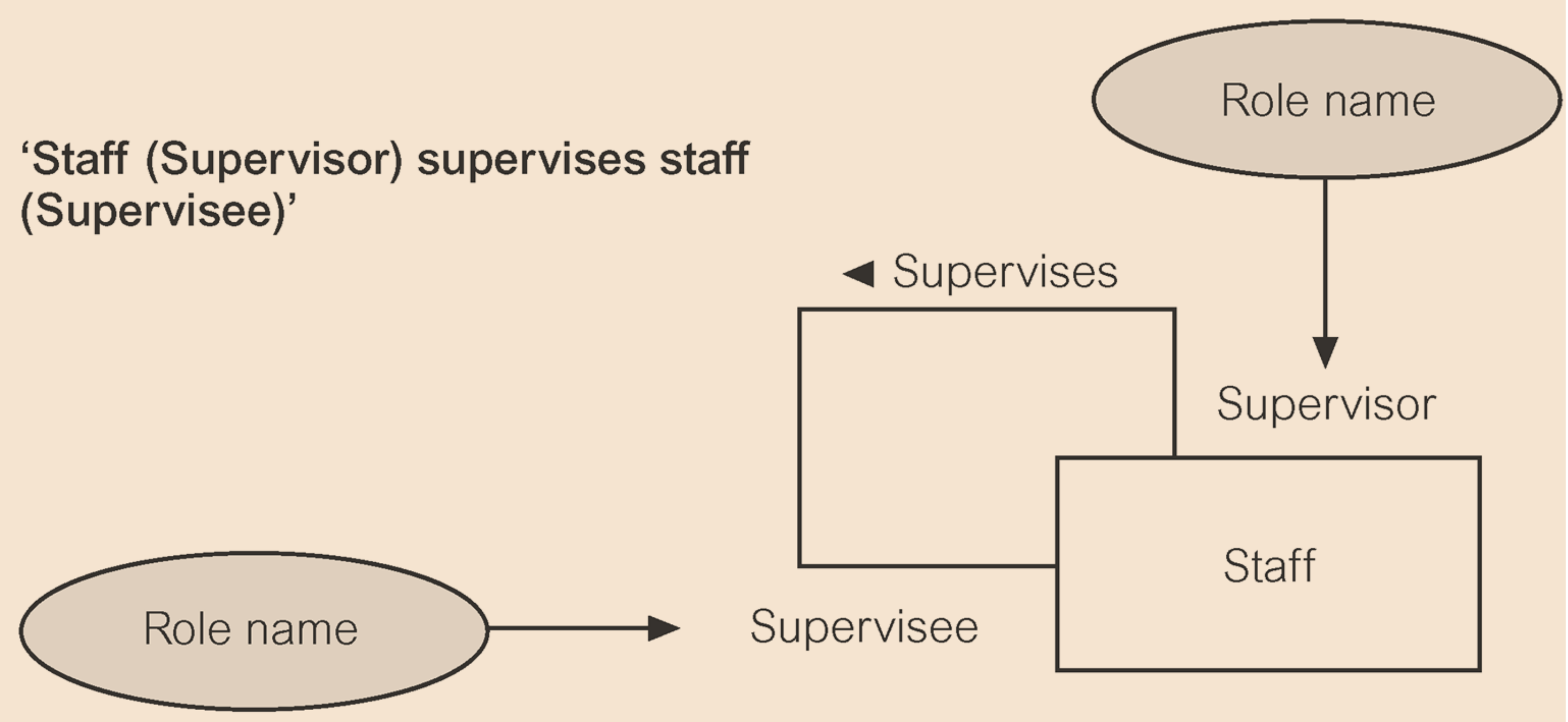
‘A solicitor arranges a bid on behalf of a buyer supported by a financial institution’

Recursive Relationship

- **Recursive Relationship (Unary relationship)**
  - A relationship type in which the same entity type participates more than once in different roles.

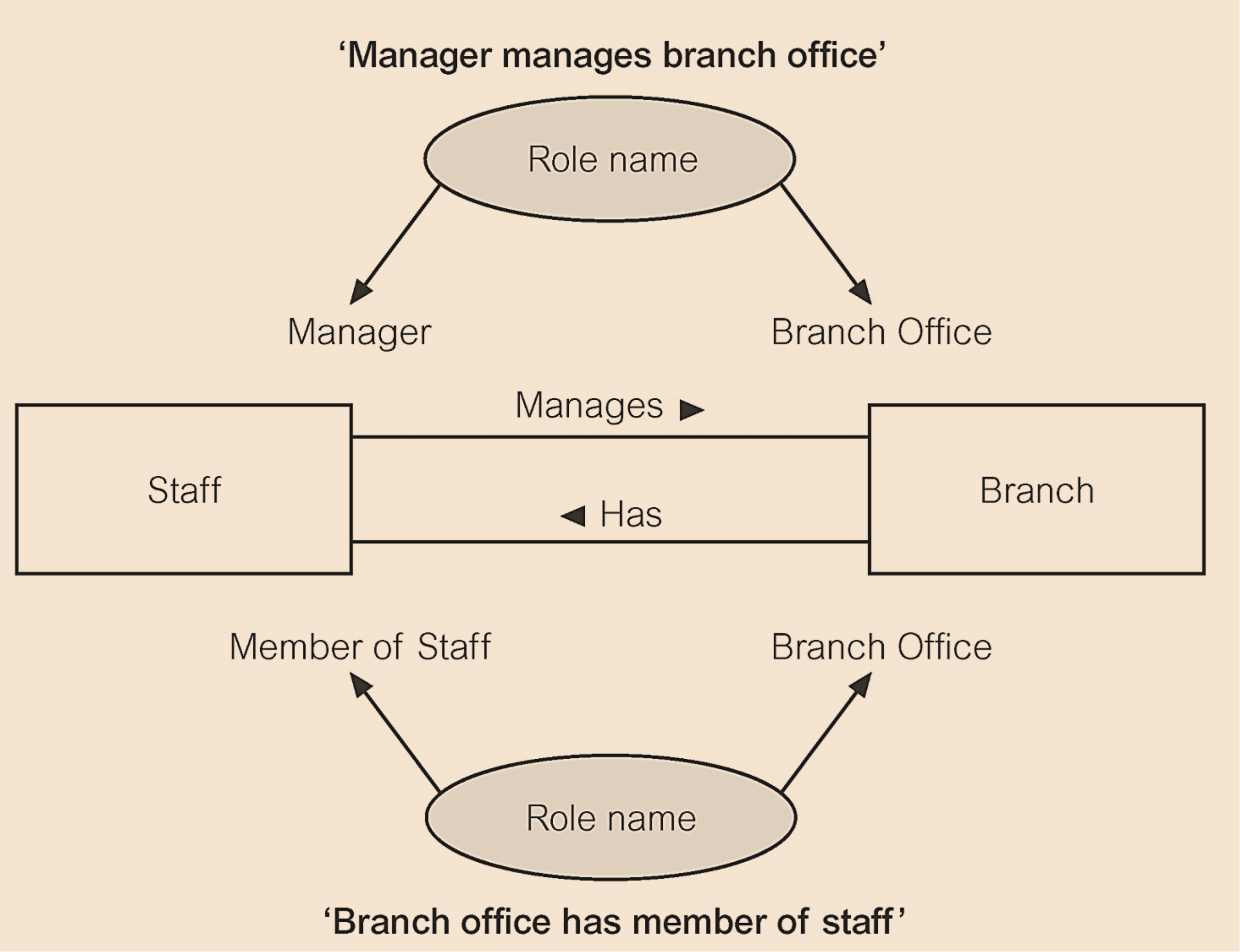
- Relationships may be given **role names** to indicate purpose that each participating entity type plays in a relationship.
  - The use of role names clarifies the purpose of each relationship.
  - Role names are usually not required if the function of the participating entities in a relationship is unambiguous.

Recursive relationship called *Supervises* with role names



Entities Associated Through 2 Distinct Relationships With Role Names

- Some entities may have multiple relationships.
- An extra line is needed for each relationship type.



Attributes

- **Attribute**
  - Property of an entity or a relationship type.
  - These will become the columns in the table.
- **Attribute Domain**
  - Set of allowable values for one or more attributes.
  - Attributes may share a domain.
    - The address attribute of the Branch, PrivateOwner and BusinessOwner entity types share the same domain of all possible addresses.
  - Domains can also be composed of domains.
    - Domain of address attribute is made up of subdomains: street, city and postCode.

Types of Attributes

- **Simple Attributes (Atomic attributes)**
  - Attribute composed of a single component with an independent existence.
  - Cannot be further subdivided into smaller components.
  - E.g. city, position, etc.
- **Composite Attributes**
  - Attribute composed of multiple components, each with an independent existence.
  - Can be further subdivided into smaller components with an independent existence of their own.
  - E.g. name, address, etc.

Types of Attributes contd..

- **Single-valued Attributes**
  - Attribute that holds a single value for each occurrence of an entity type.
  - E.g. name, currentAddress, etc.

- **Multi-valued Attributes**
  - Attribute that holds multiple values for each occurrence of an entity type.
  - E.g. email, phone, etc.

Types of Attributes contd..

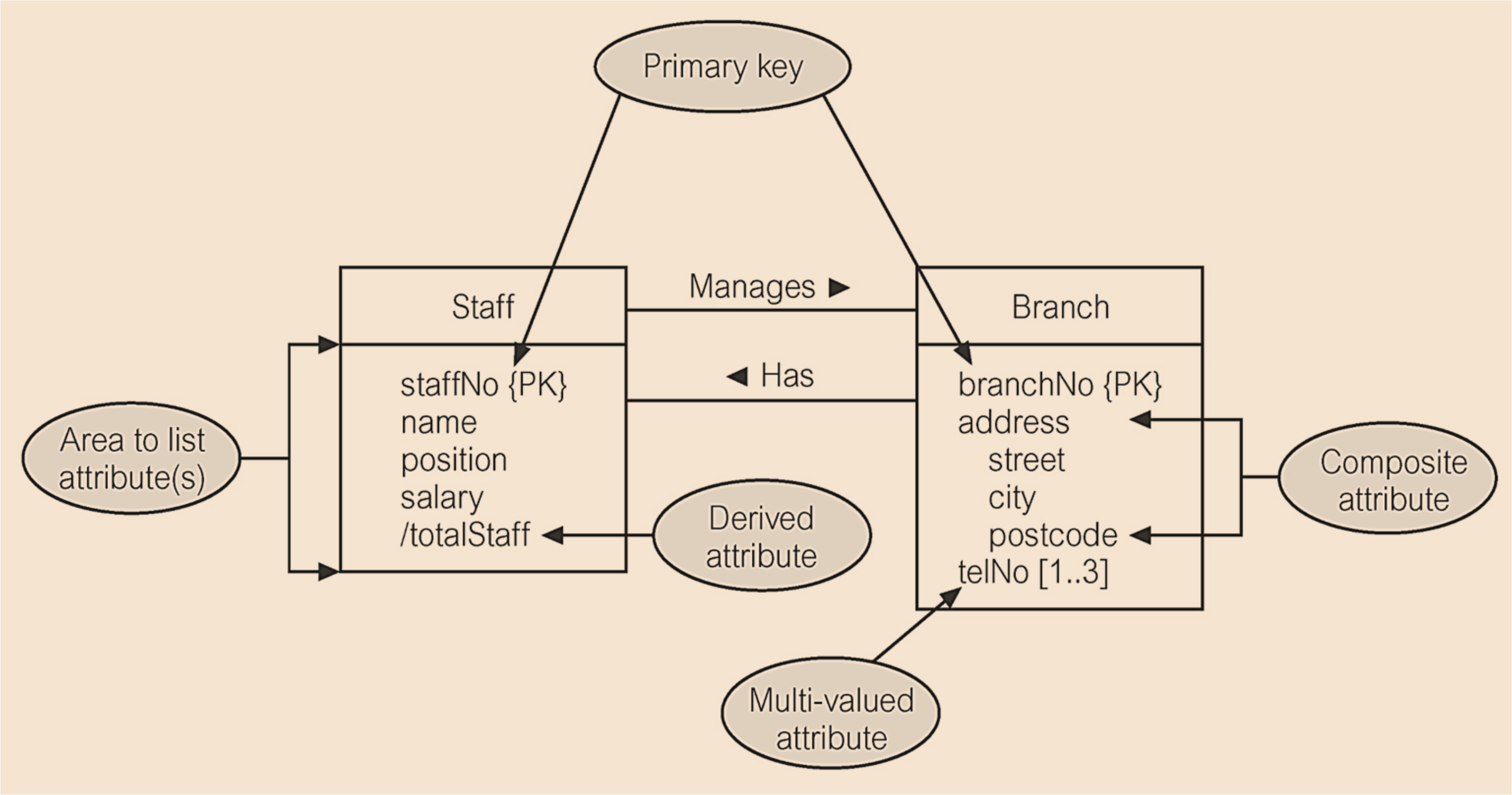
- **Derived Attributes**
  - Attribute that represents a value that is derivable from value of a related attribute, or set of attributes, not necessarily in the same entity type.
  - Derived attributes are never stored in a table.
  - E.g. Age, profit, loss, etc.

Concept of Keys in ERD

- **Candidate Key**
  - Minimal set of attributes that uniquely identifies each occurrence of an entity type.
- **Primary Key**
  - Candidate key selected to uniquely identify each occurrence of an entity type.
- **Composite Key**
  - A candidate key that consists of two or more attributes.

ER Diagram of Staff and Branch Entities and their Attributes

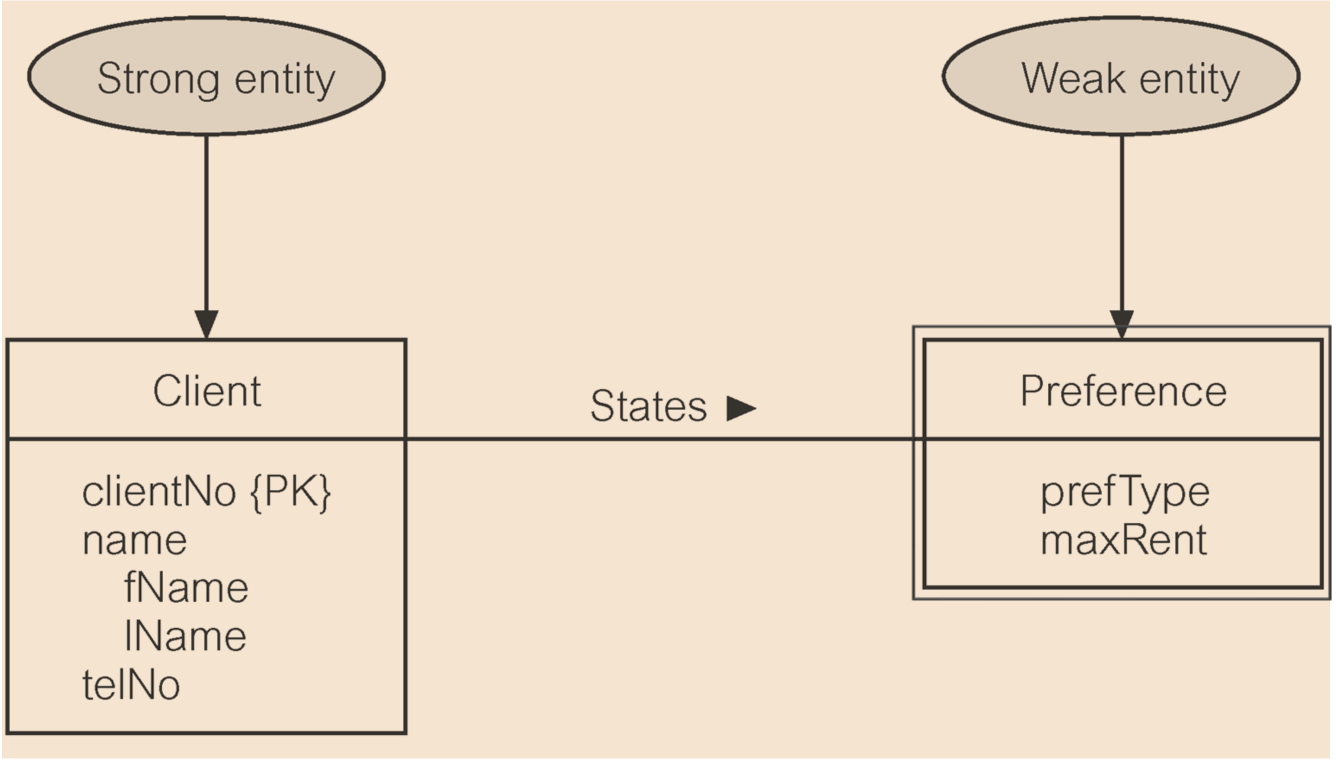
Additional tags that can be used include partial PK {PPK} (when an attribute forms part of a composite PK) and alternate key {AK}.



Classification of Entity Types

- **Strong Entity Type (parent/owner/dominant)**
  - Entity type that is *not* existence-dependent on some other entity type.
  - Each entity occurrence is uniquely identifiable using the PK attribute(s) of that entity type.
  - E.g. Client, Customer, Textbook, etc.
- **Weak Entity Type (child/dependent/subordinate)**
  - Entity type that is existence-dependent on some other entity type.
  - Each entity occurrence cannot be uniquely identified using only the attributes associated with that entity type.
  - E.g. Preference, Order, Edition, etc.

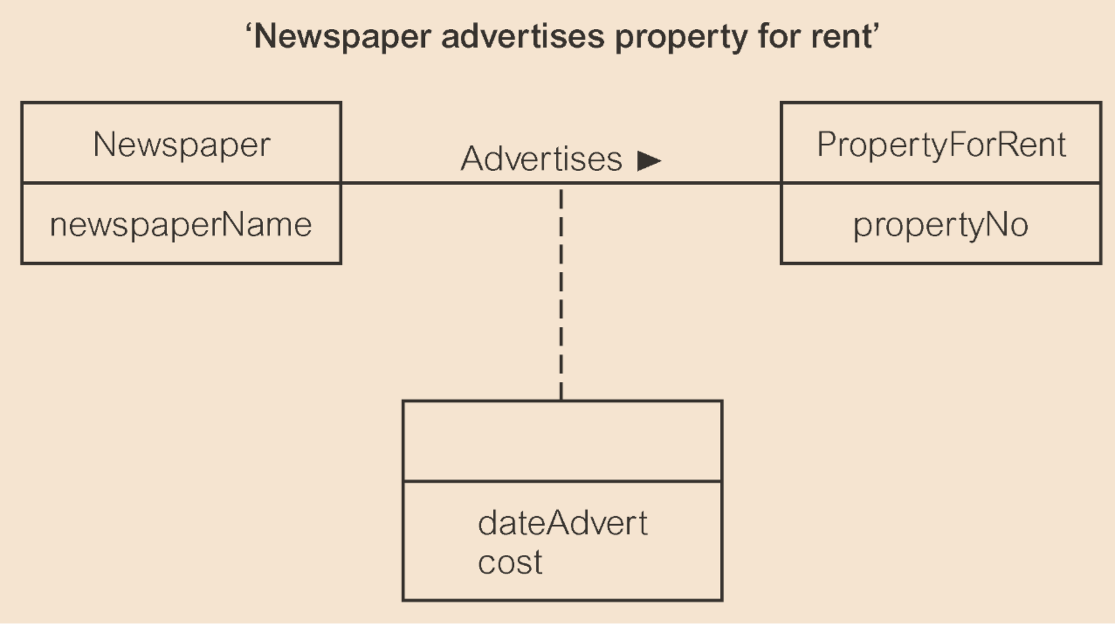
Strong entity type called *Client* and weak entity type called *Preference*



Attributes on Relationships

Relationship called *Advertises* with attributes

To record the date when the property was advertised and the cost of this advertisement, we associate this information with the *Advertises* relationship as attributes called ***dateAdvert*** and ***cost***, rather than with the *Newspaper* or the *PropertyForRent* entities.



Main Point

The E-R model employs three basic concepts: entity types, relationship types, and attributes.

Science & Technology of Consciousness: On the subjective side, Rishi (knower), Devata (process of knowing) and Chhandas (known) are the three basic qualities that structure all of creation.

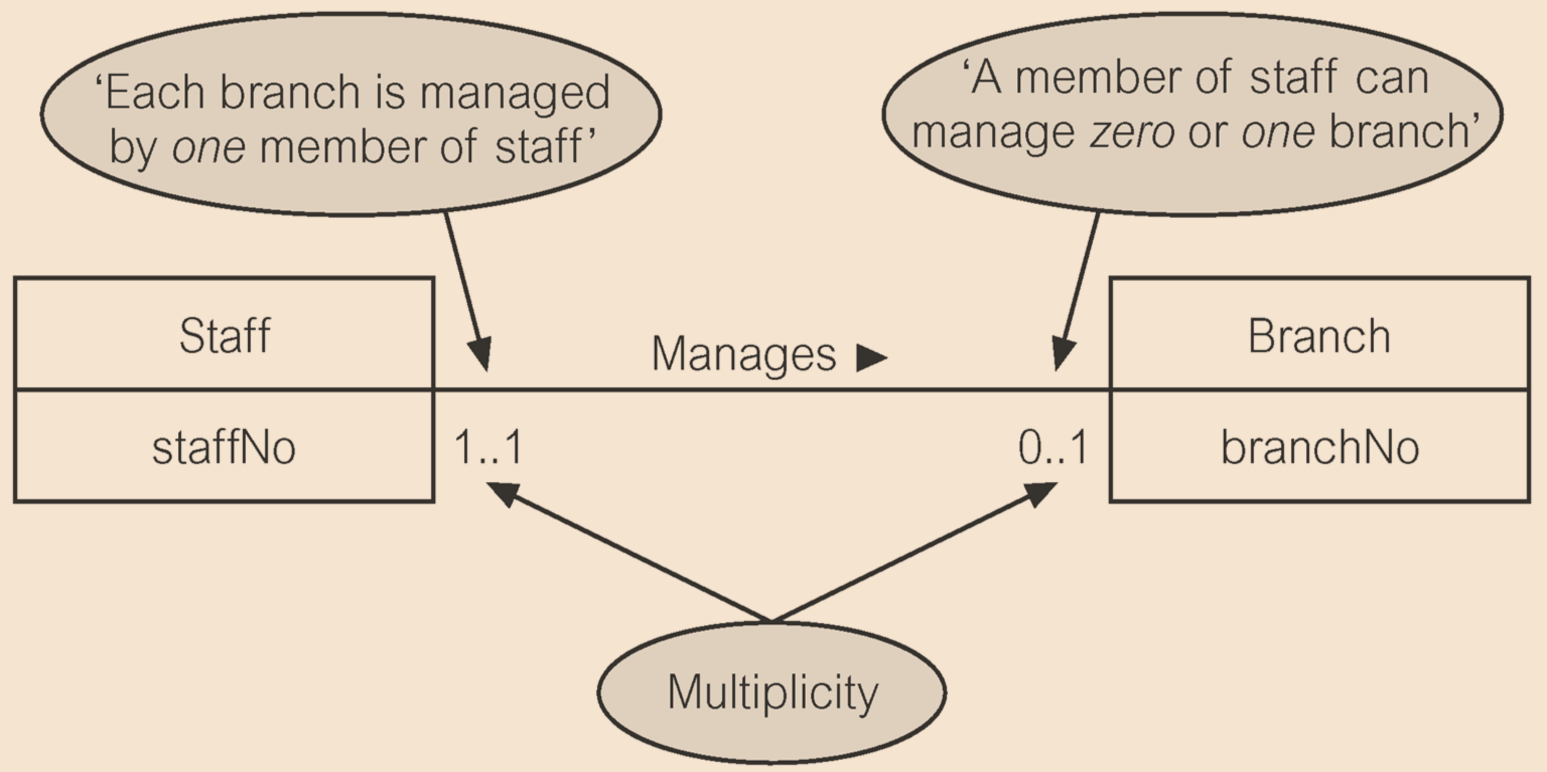
Structural Constraints

- Constraints that may be placed on entity types that participate in a relationship.
- The constraints should reflect the restrictions on the relationships as perceived in the "real world."
- Main type of constraint on relationships is called **multiplicity**.
  - Represents policies (called business rules) established by user or company.

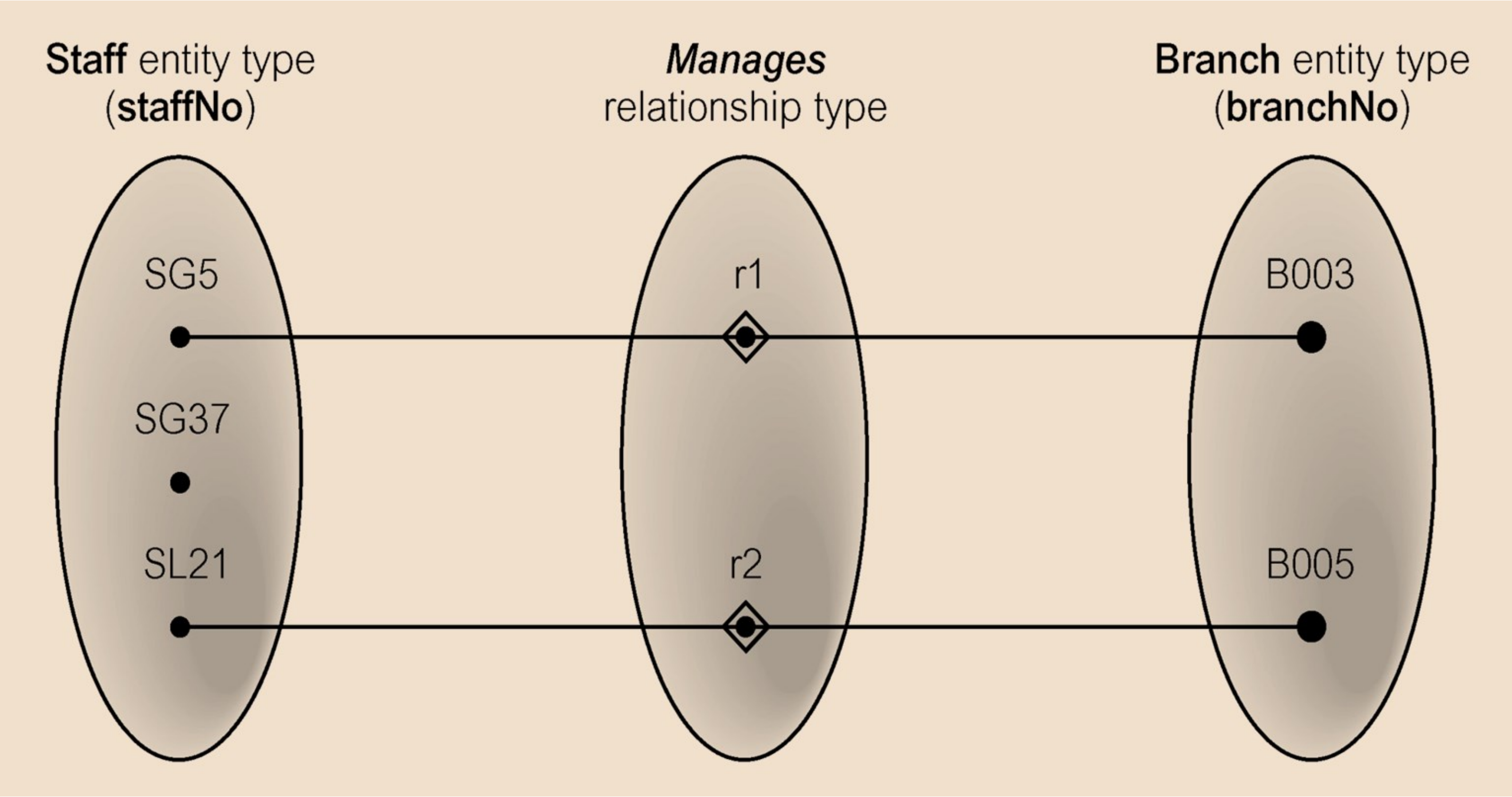
Multiplicity

- **Multiplicity** - number (or range) of possible occurrences of an entity type that may relate to a single occurrence of an associated entity type through a particular relationship.
- Determining the multiplicity normally requires examining the precise relationships between the data given in an enterprise constraint using sample data.
- Sample data must be a true representation of all the data being modeled.

Multiplicity of *Staff Manages Branch* relationship


'Each branch is managed by *one* member of staff'
'A member of staff can manage *zero* or *one* branch'
Staff
staffNo
Manages ►
1..1
0..1
Branch
branchNo
Multiplicity

Semantic net of *Staff Manages Branch* relationship type



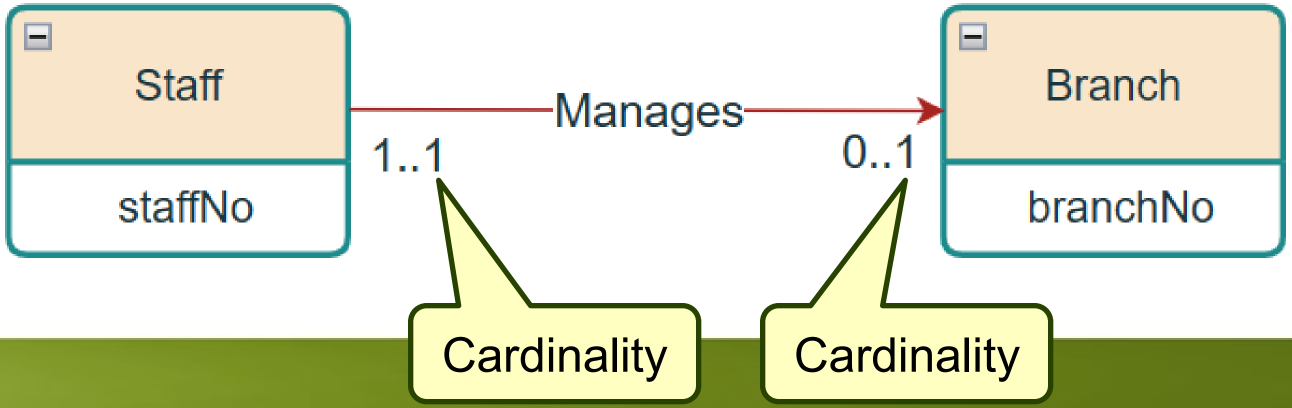
Cardinality & Participation Constraints

- **Multiplicity** actually consists of two separate constraints known as **cardinality** and **participation**.
- **Cardinality**
  - Describes maximum number of possible relationship occurrences for an entity participating in a given relationship type.
- **Participation**
  - Determines whether all or only some entity occurrences participate in a relationship.

Cardinality

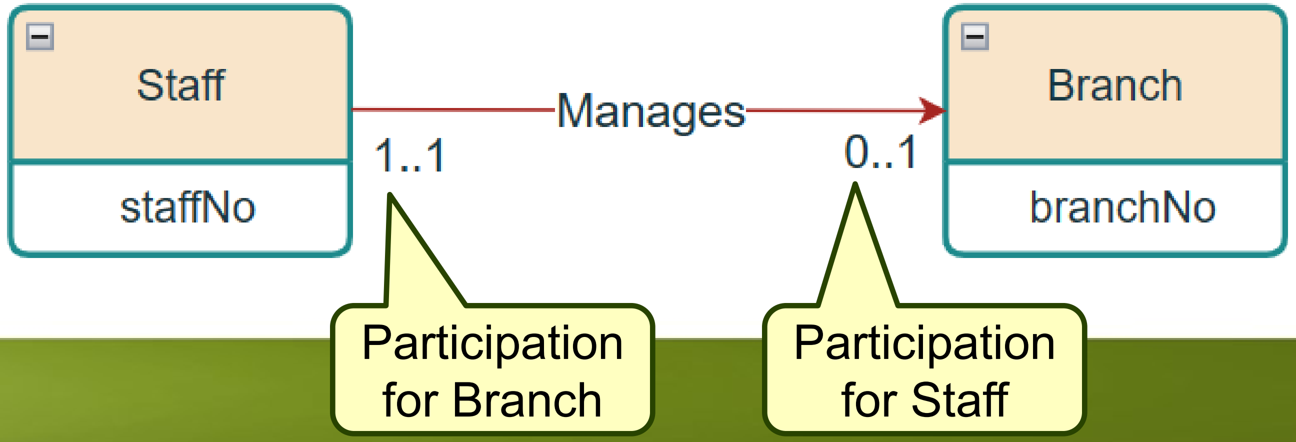
- The cardinality of a relationship appears as the ***maximum*** value for the multiplicity ranges on either side of the relationship.

For example, the *Manages* relationship has a one-to-one (1:1) cardinality and this is represented by multiplicity ranges with a maximum value of 1 on both sides of the relationship.



Participation

- The participation of entities in a relationship appears as the ***minimum*** values for the multiplicity ranges on either side of the relationship.
- Participation for a given entity in a relationship is represented by the minimum value on the ***opposite*** side of the relationship; that is the minimum value for the multiplicity beside the related entity.



Participation for Branch

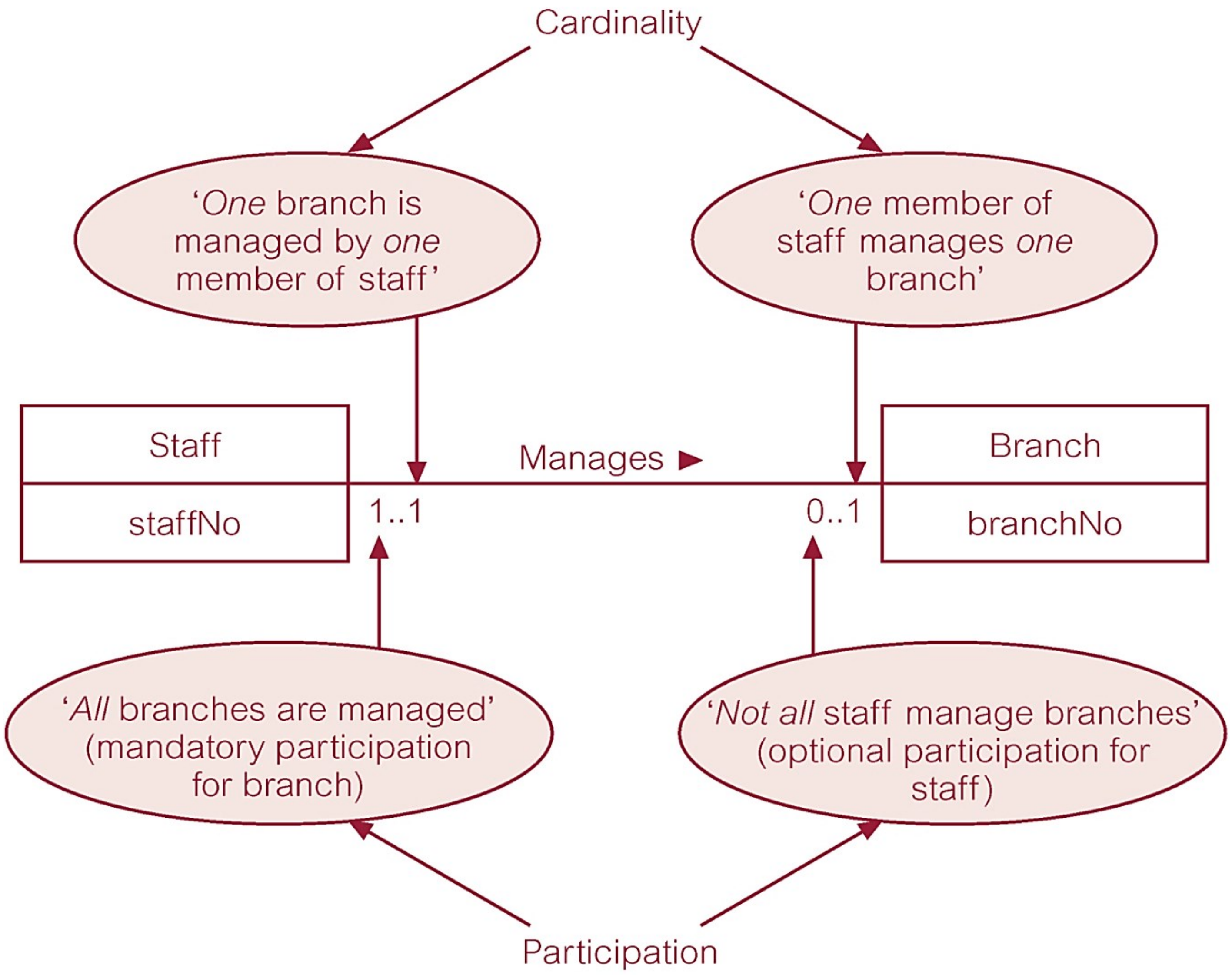
Participation for Staff

Types of Participation Constraints

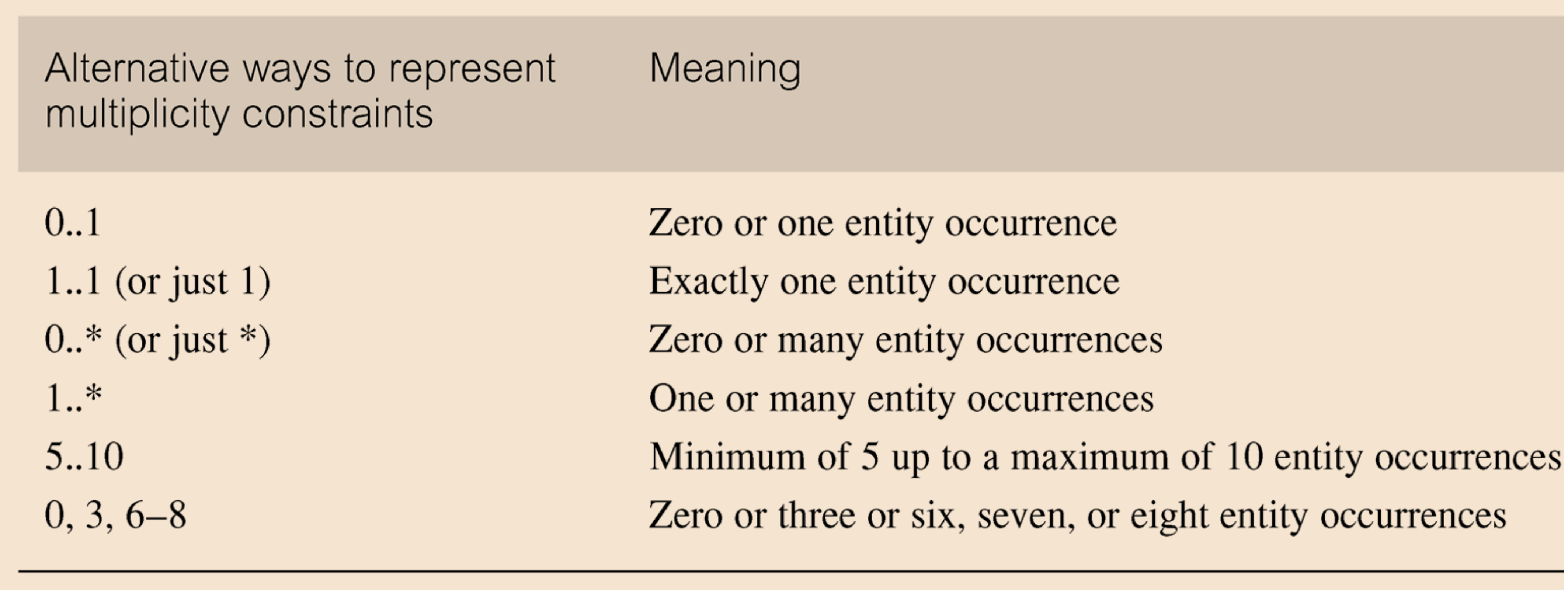
- **Mandatory participation**
  - All entity occurrences are involved in a particular relationship.
  - Represented as minimum value of 1.

- **Optional participation**
  - Only some entity occurrences are involved in a particular relationship.
  - Represented as minimum value of 0.

- **Optional participation for the Staff entity** in the *Manages* relationship is shown as a minimum value of 0 for the multiplicity beside the Branch entity.
- **Mandatory participation for the Branch entity** in the *Manages* relationship is shown as a minimum value of 1 for the multiplicity beside the Staff entity.


Cardinality
'One branch is managed by one member of staff'
'One member of staff manages one branch'
Staff
staffNo
Manages ►
1..1
0..1
Branch
branchNo
'All branches are managed' (mandatory participation for branch)
'Not all staff manage branches' (optional participation for staff)
Participation

Summary of Multiplicity Constraints



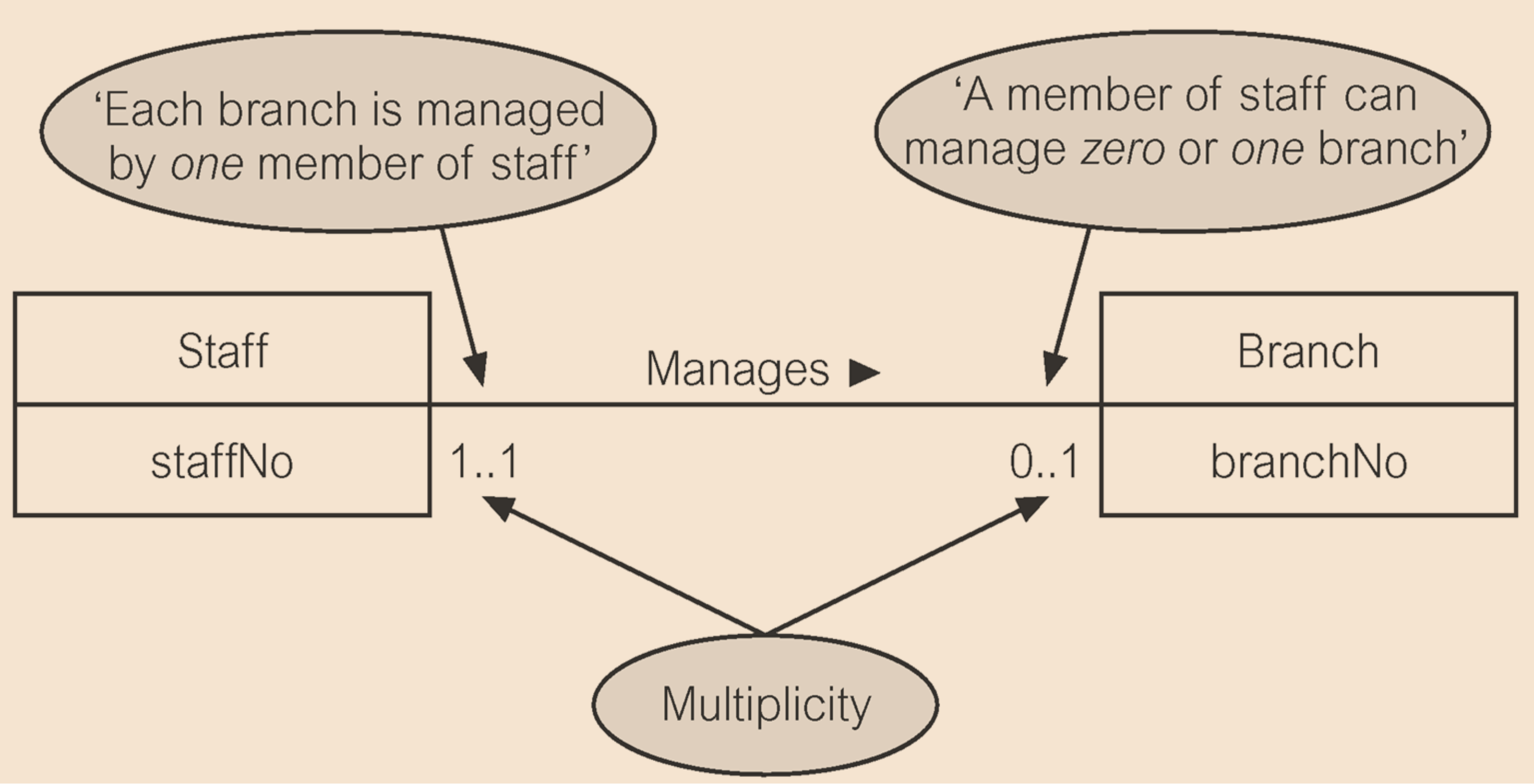
Alternative ways to represent multiplicity constraints	Meaning
0..1	Zero or one entity occurrence
1..1 (or just 1)	Exactly one entity occurrence
0..* (or just *)	Zero or many entity occurrences
1..*	One or many entity occurrences
5..10	Minimum of 5 up to a maximum of 10 entity occurrences
0, 3, 6–8	Zero or three or six, seven, or eight entity occurrences

Binary Relationship Types

- The most common degree for relationships is binary.

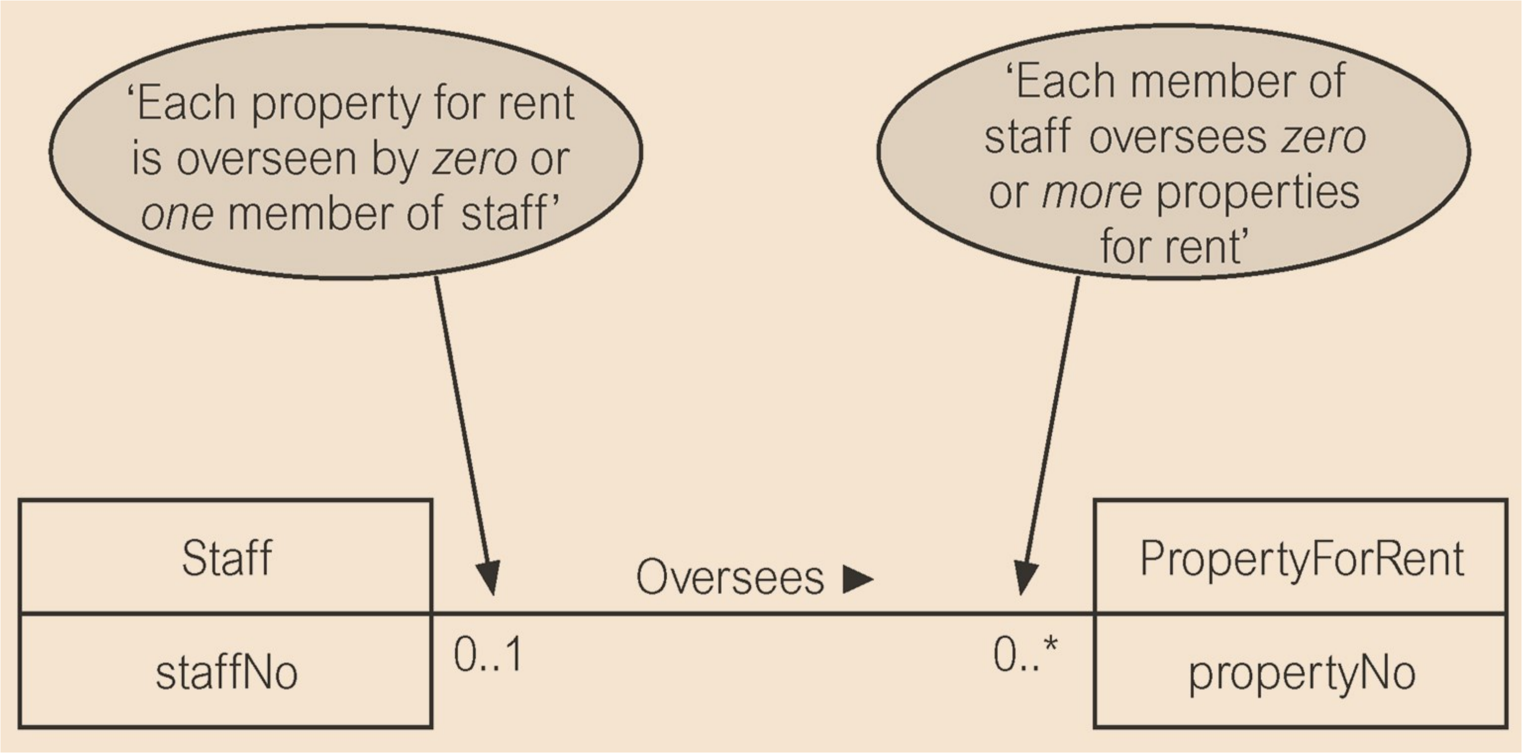
- Binary relationships are generally referred to as being:
  - **one-to-one (1:1)**
  - **one-to-many (1:*)**
  - **many-to-many (*:*)**

Staff *Manages* Branch - 1:1 relationship



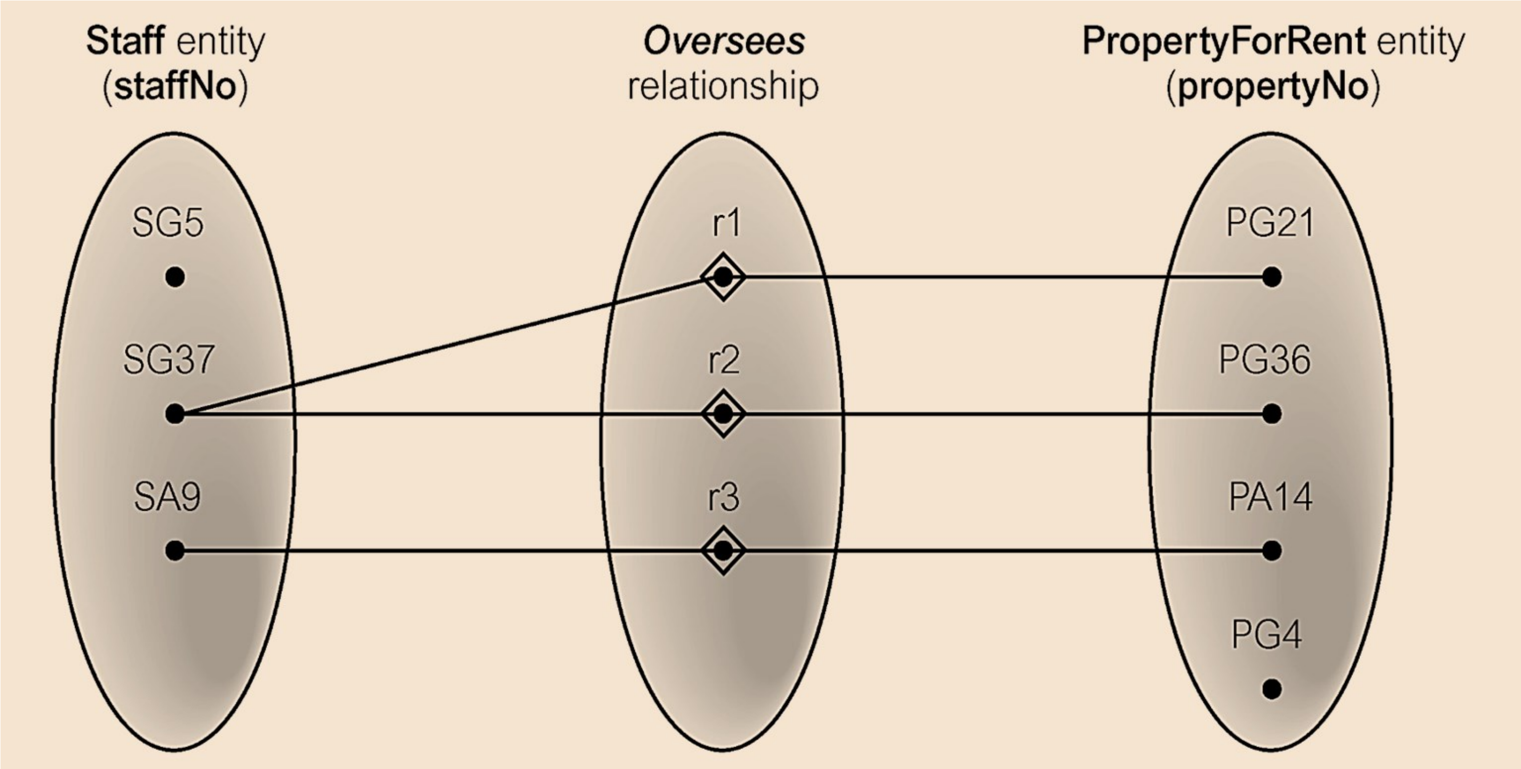
‘Each branch is managed by *one* member of staff’
‘A member of staff can manage *zero* or *one* branch’
Staff
staffNo
Manages ▶
1..1
0..1
Branch
branchNo
Multiplicity

Staff *Oversees* PropertyForRent – 1:* relationship

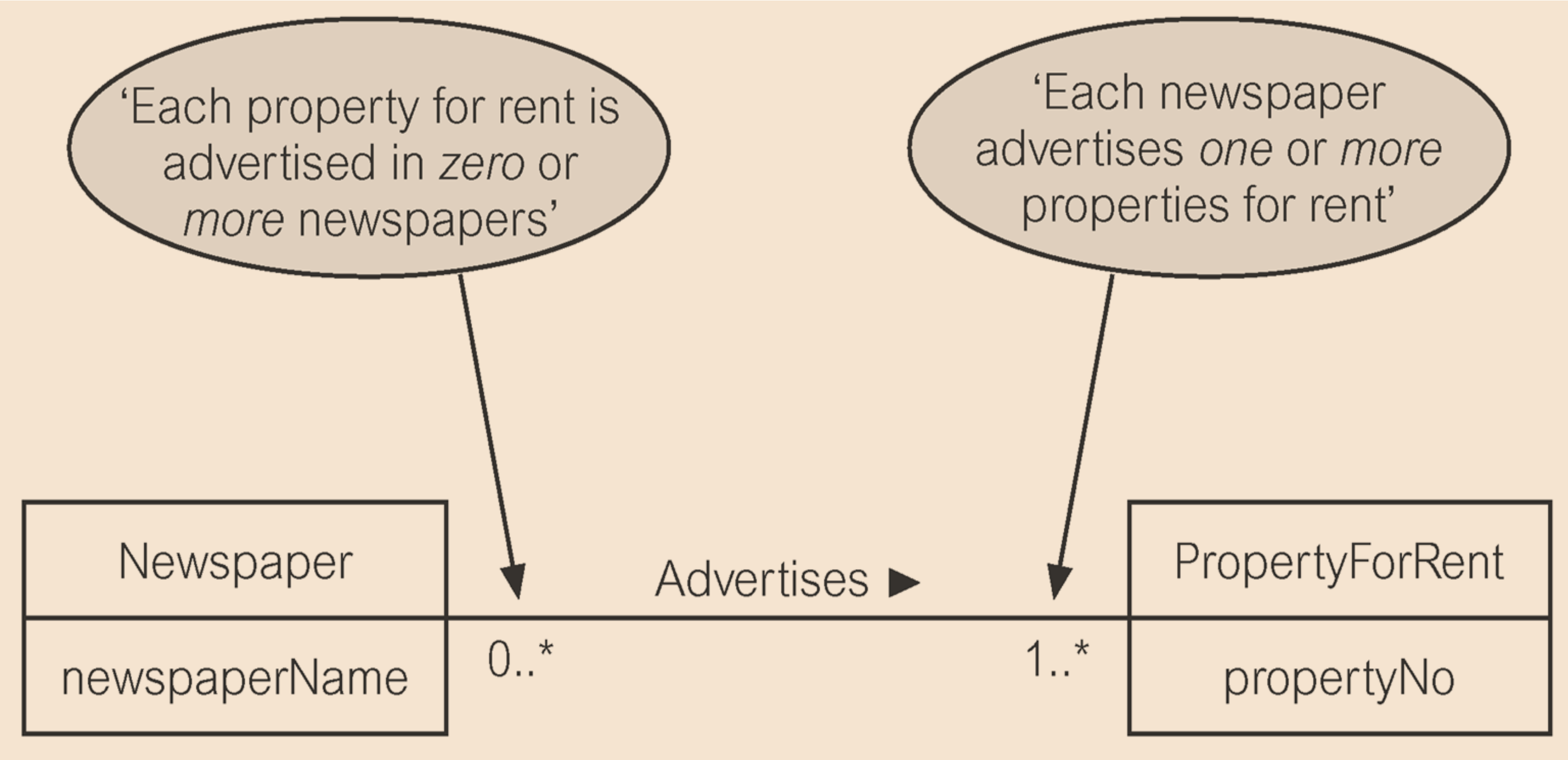


'Each property for rent is overseen by *zero* or *one* member of staff'
'Each member of staff oversees *zero* or *more* properties for rent'
Staff
staffNo
Oversees ▶
0..1
0..*
PropertyForRent
propertyNo

Semantic net of Staff *Oversees* PropertyForRent relationship type

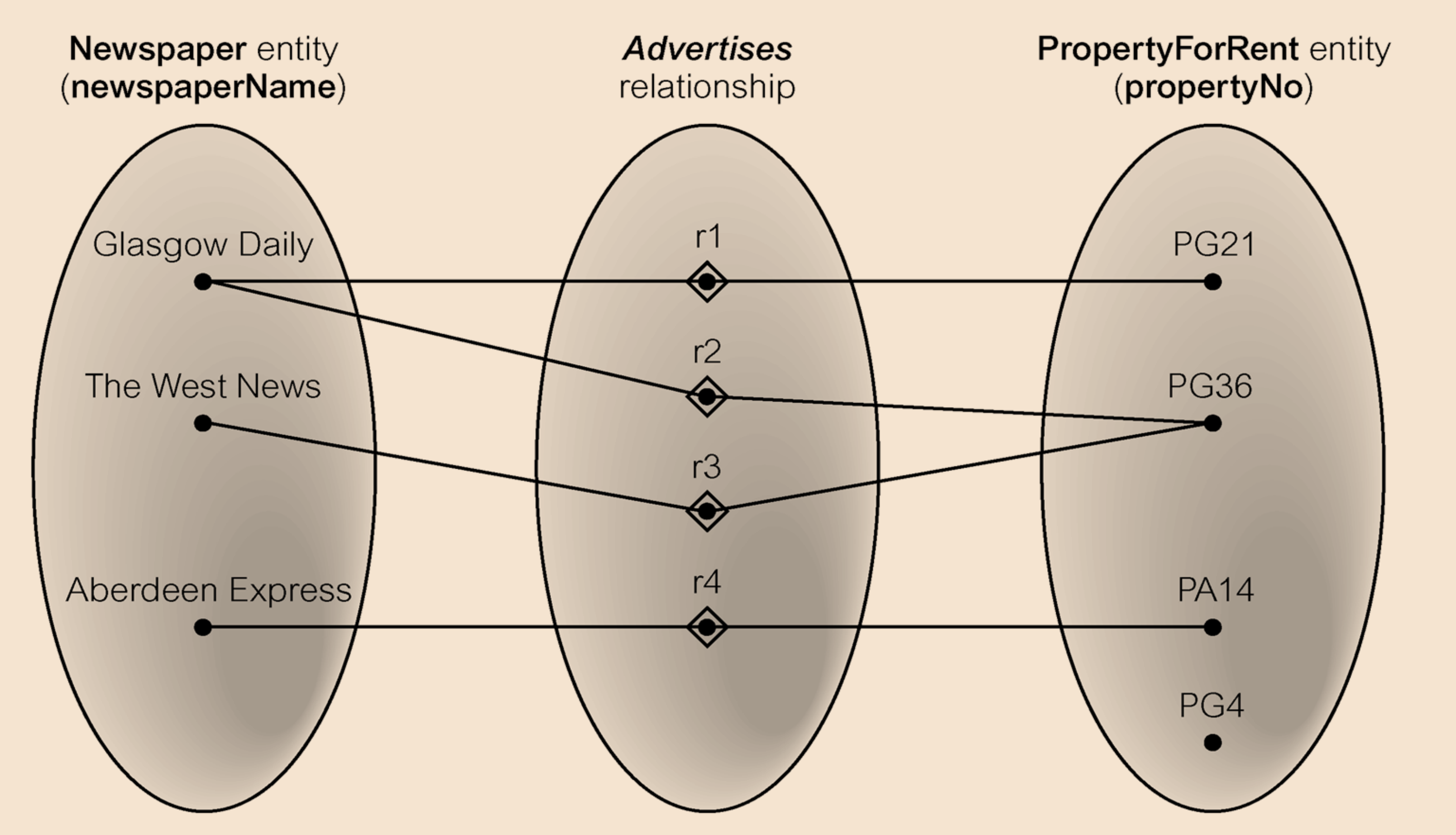


Newspaper *Advertises* PropertyForRent *:* relationship



‘Each property for rent is advertised in *zero* or *more* newspapers’
‘Each newspaper advertises *one* or *more* properties for rent’
Newspaper
newspaperName
Advertises ▶
0..*
1..*
PropertyForRent
propertyNo

Semantic net of Newspaper *Advertises* PropertyForRent relationship type



Problems with ER Models

- Problems may arise when designing a conceptual data model; these problems are called as ***connection traps***.

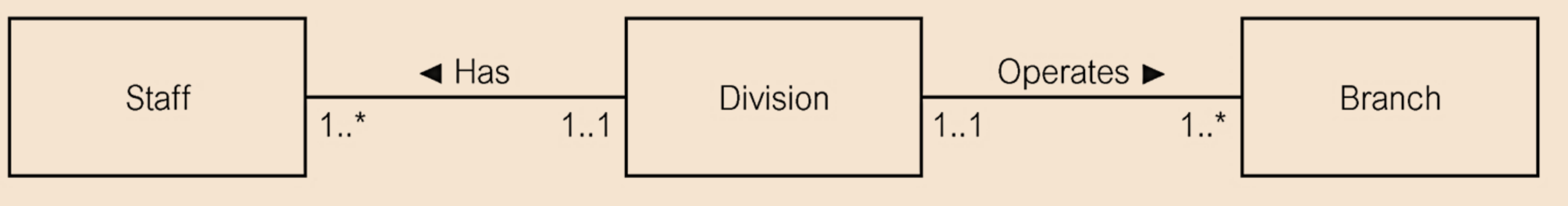
- Often occur due to a misinterpretation of the meaning of certain relationships.

- Two main types of connection traps are called ***fan traps*** and ***chasm traps***.

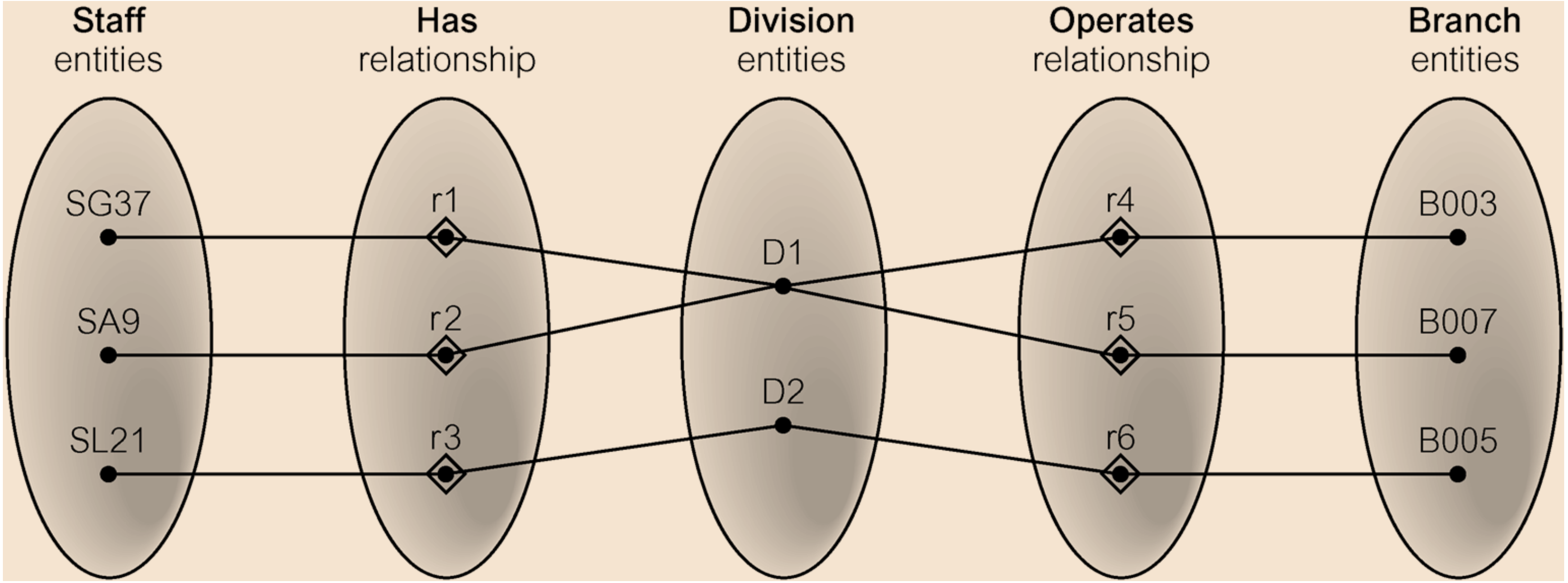
Problems with ER Models

1. Fan Trap

- Where an ER model represents a relationship between entity types, but pathway between certain entity occurrences is ambiguous.
- A fan trap may exist where two or more 1:* relationships fan out from the same entity.



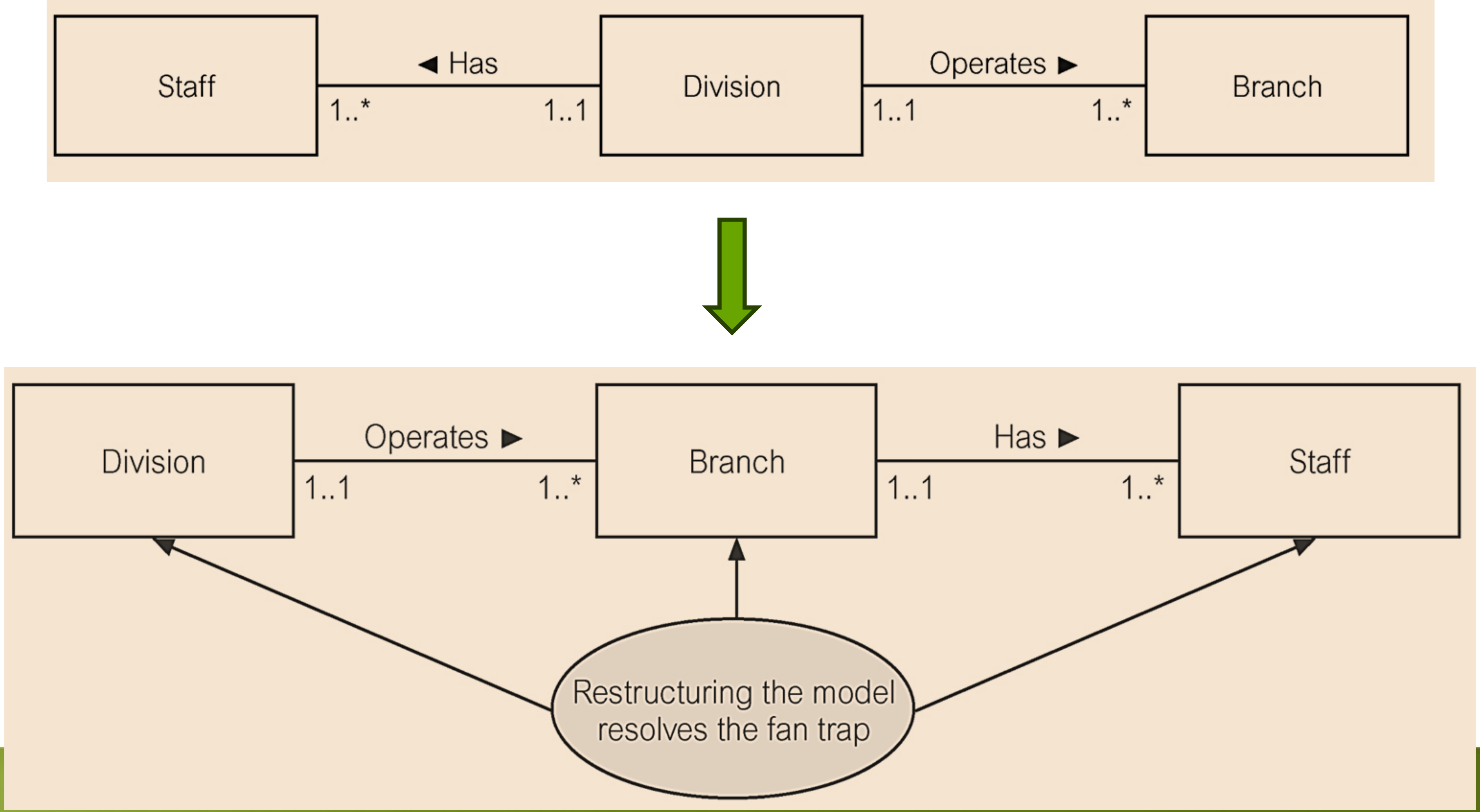
Semantic Net of ER Model with Fan Trap



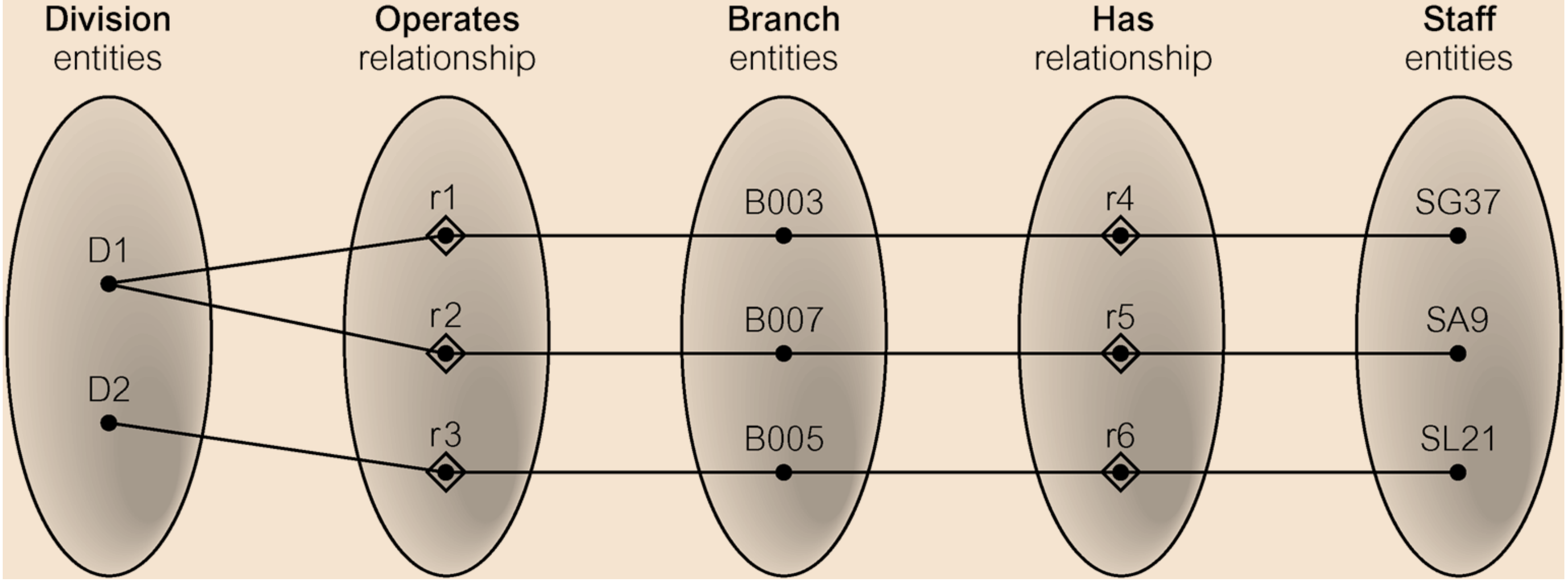
At which branch office does staff number SG37 work?

- Inability to answer this question is the result of a fan trap associated with the misrepresentation of the correct relationships between the Staff, Division, and Branch entities.

Restructuring ER Model to Remove Fan Trap



Semantic Net of Restructured ER Model with Fan Trap Removed

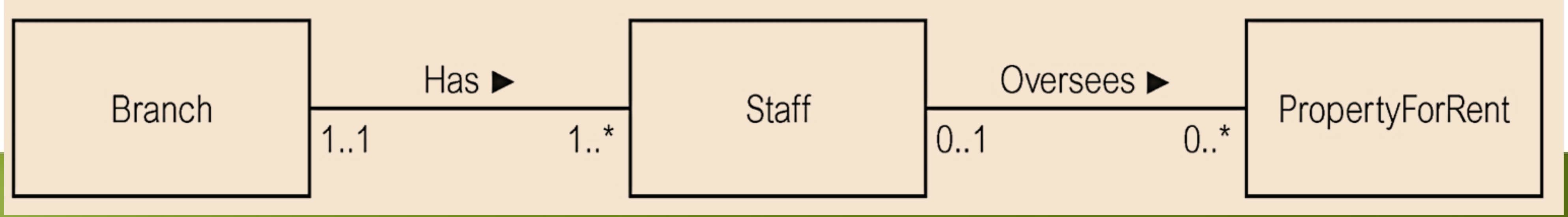


- SG37 works at branch B003.

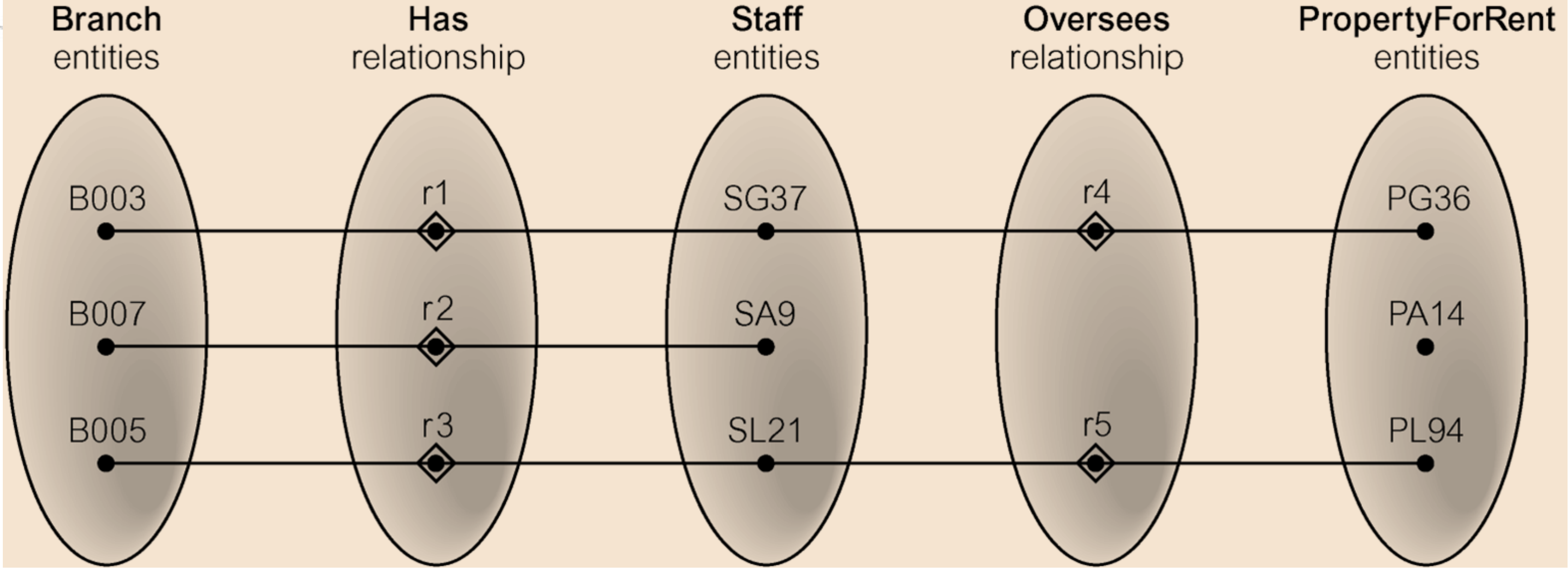
Problems with ER Models

2. Chasm Trap

- Where an ER model suggests the existence of a relationship between entity types, but pathway does not exist between certain entity occurrences.
- A chasm trap may occur where there are one or more relationships with a minimum multiplicity of zero (i.e. optional participation) forming part of the pathway between related entities.



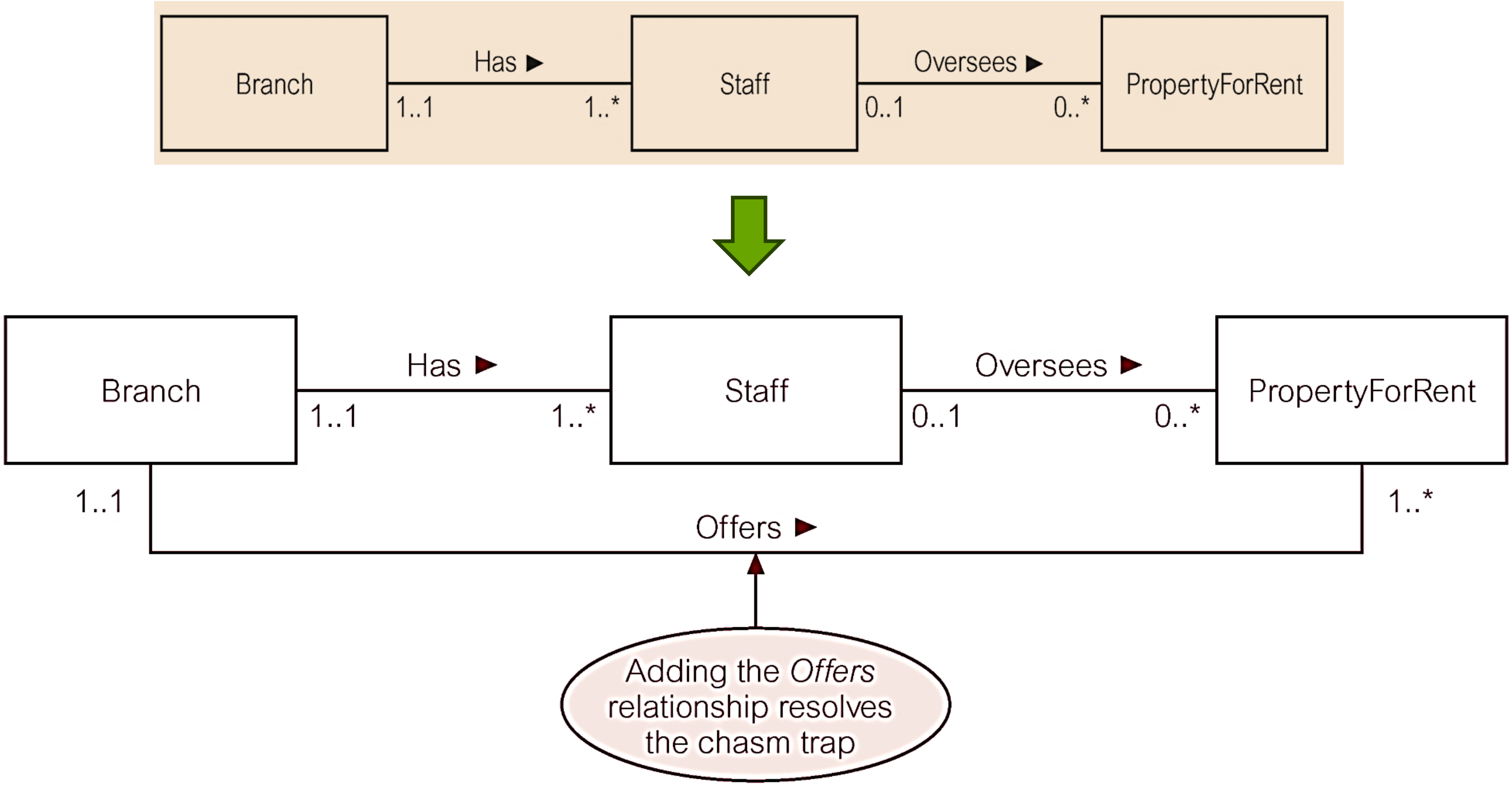
Semantic Net of ER Model with Chasm Trap



At which branch office is property PA14 available?

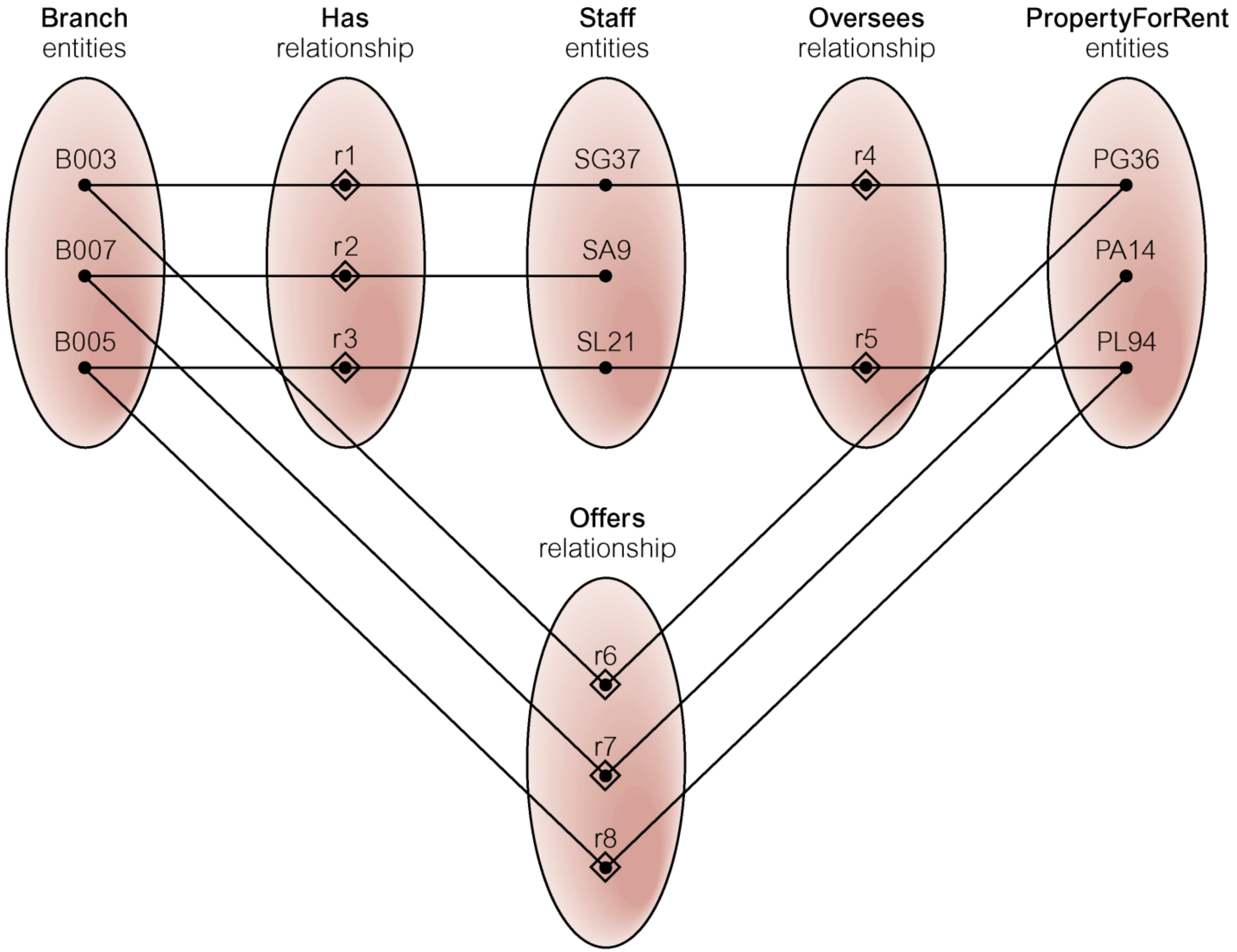
- The inability to answer this question is considered to be a loss of information (as we know a property must be available at a branch), and is the result of a chasm trap.

ER Model Restructured To Remove Chasm Trap



Adding the *Offers* relationship resolves the chasm trap

Semantic Net of Restructured ER Model with Chasm Trap Removed

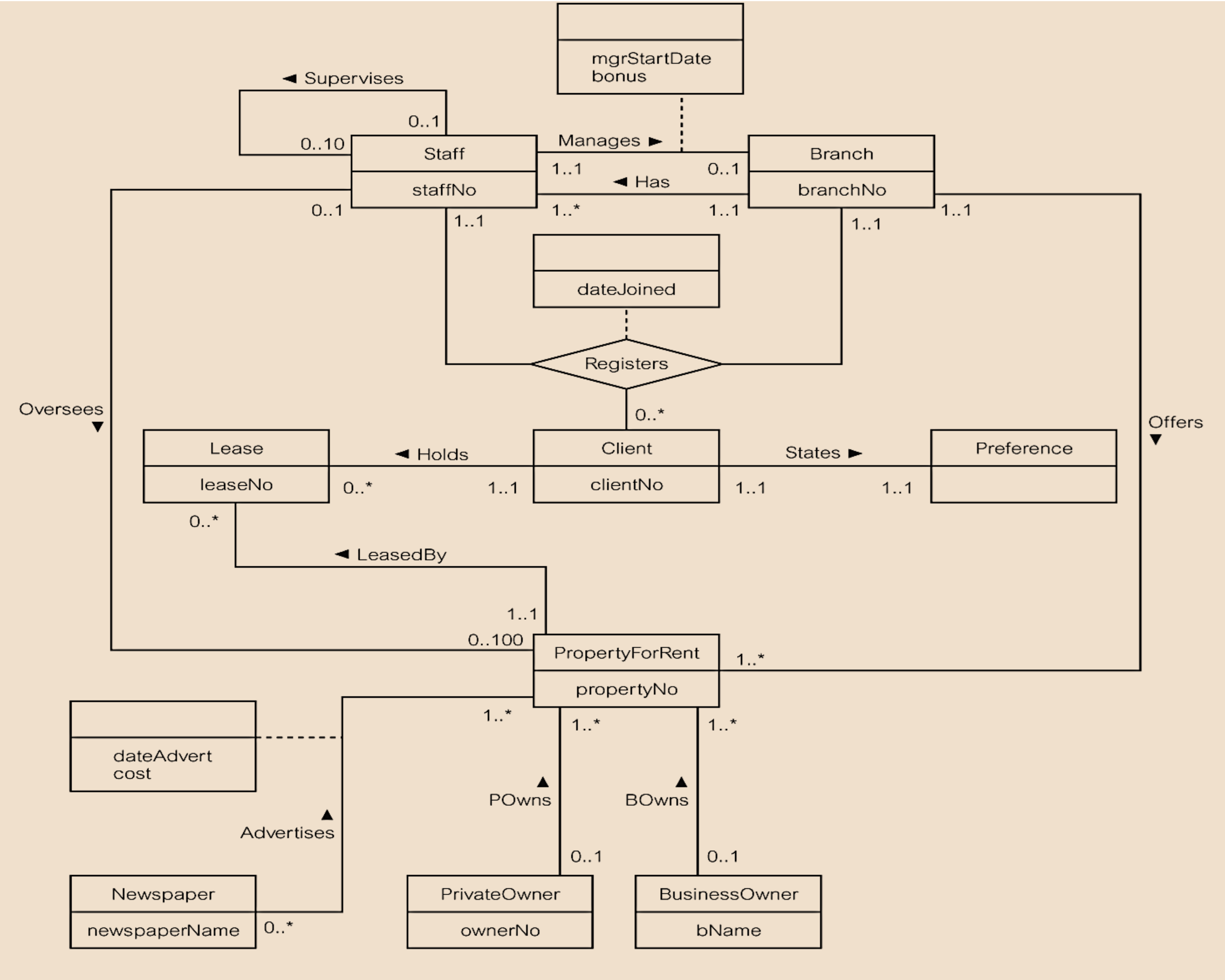


Main Point

The overall logical structure of a database can be expressed graphically by an E-R diagram which gives a holistic overview of the data that will be in the database. It is important that each relationship be understood in terms of the whole diagram; otherwise connection traps may occur.

Science & Technology of Consciousness: A graphical technique employed by Vedic science is the unified field chart which gives a holistic overview of a discipline and links all knowledge with the Self.

ERD of Branch user views of DreamHome



UNITY CHART

CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE:

E-R Modeling as a Database Design Technique

1. The E-R model employs three basic concepts: entity sets, relationship sets, and attributes.
2. An ER diagram is a graphic technique for expressing the wholeness of a database. Entities are very slippery things; one person’s entity may be another person’s relationship!

3. Transcendental consciousness is the experience of the simplest and most abstract state of awareness which underlies all states of greater excitation.
4. Impulses within the Transcendental Field: Nature accomplishes what it needs by having its impulses in the transcendental field be as efficient as possible.
5. **Wholeness moving within itself:** In unity consciousness one experiences that all layers of the universe are only different expressions of the same infinite field of pure consciousness.