Chapter 12

Entity-Relationship Modeling

Objectives for This Lecture

- Basic concepts associated with ER model.
- Diagrammatic technique for displaying ER model using Unified Modeling Language (UML).
- How to identify and resolve problems with ER models called connection traps.

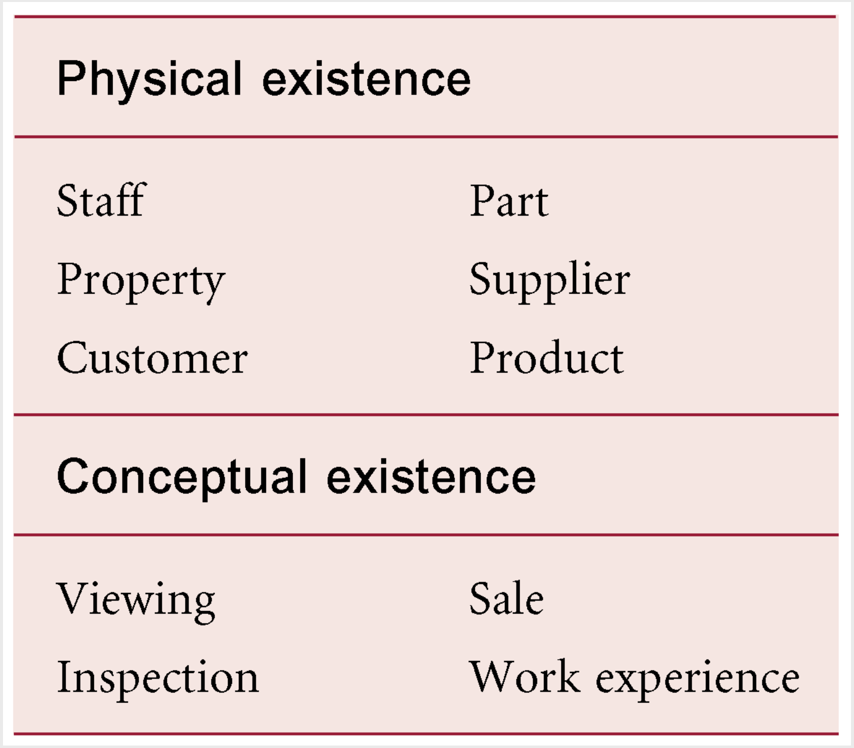
Concepts of the ER Model

- Entity types
- Relationship types
- Attributes

Entity Type

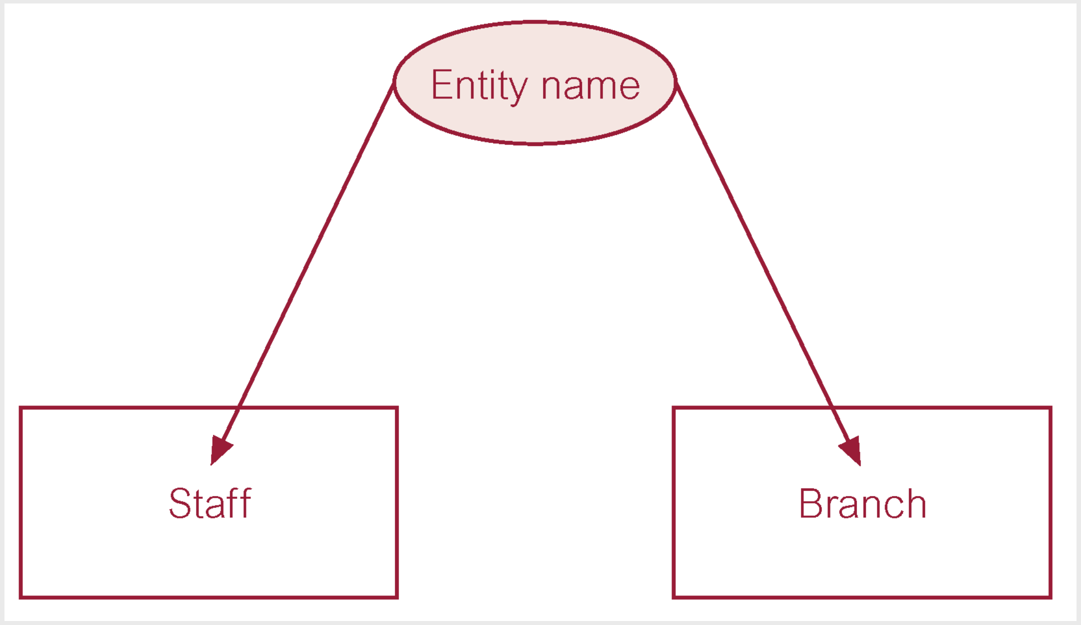
- **Entity: Group of objects with same properties, identified by enterprise as having an independent existence.**

Examples of Entity Types



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

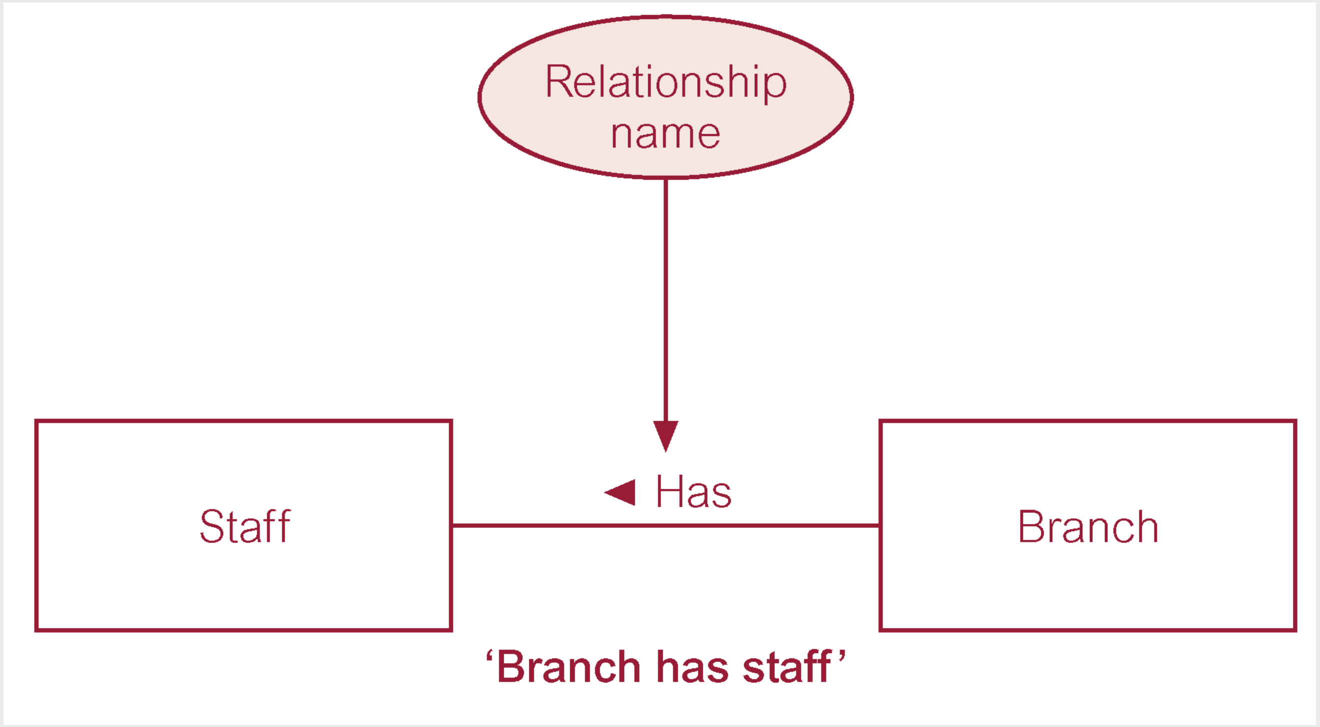
ER diagram of Staff and Branch entity types



Relationship Types

- Relationship: A meaningful association among entity types.
- Represented in relations by foreign keys

ER diagram of Branch *Has* Staff relationship

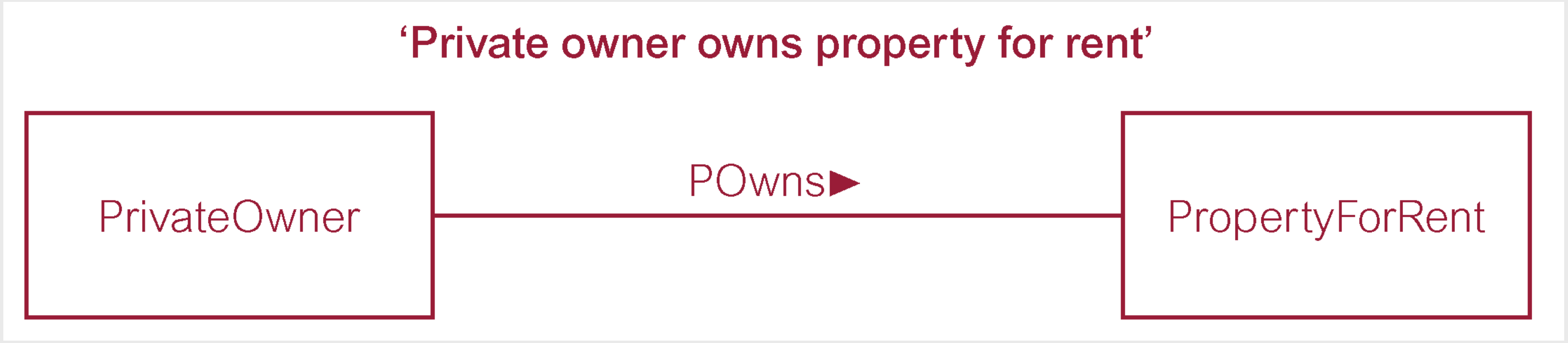


Relationship Types

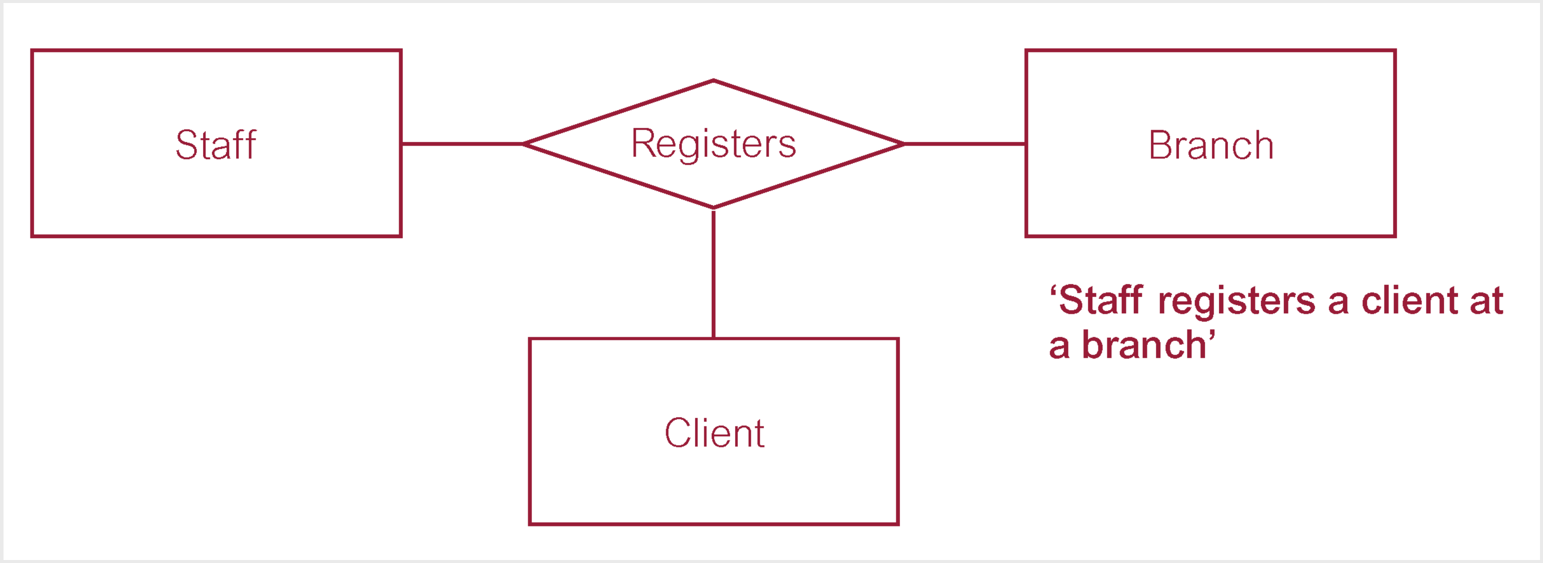
- Degree of a Relationship
  - Number of participating entities in relationship.

- Relationship of degree :
  - two is binary
  - three is ternary
  - four is quaternary.

Binary relationship called *POwns*

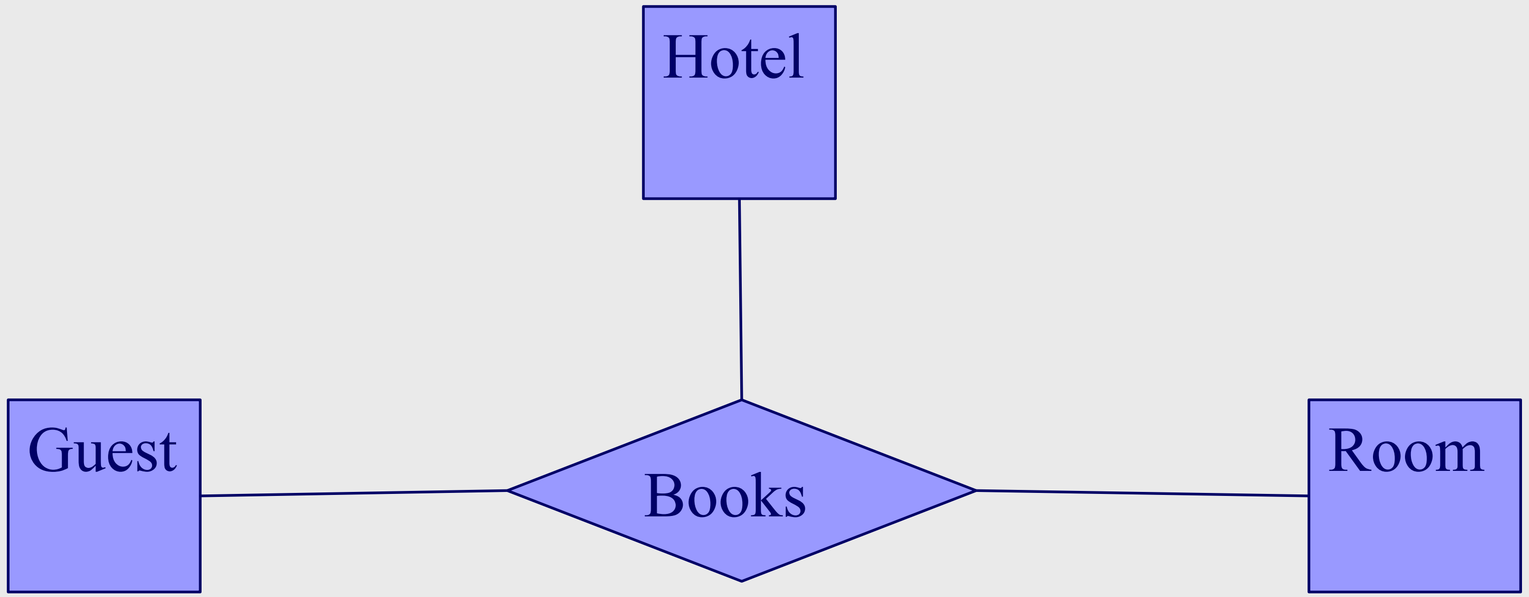


Ternary relationship called *Registers*

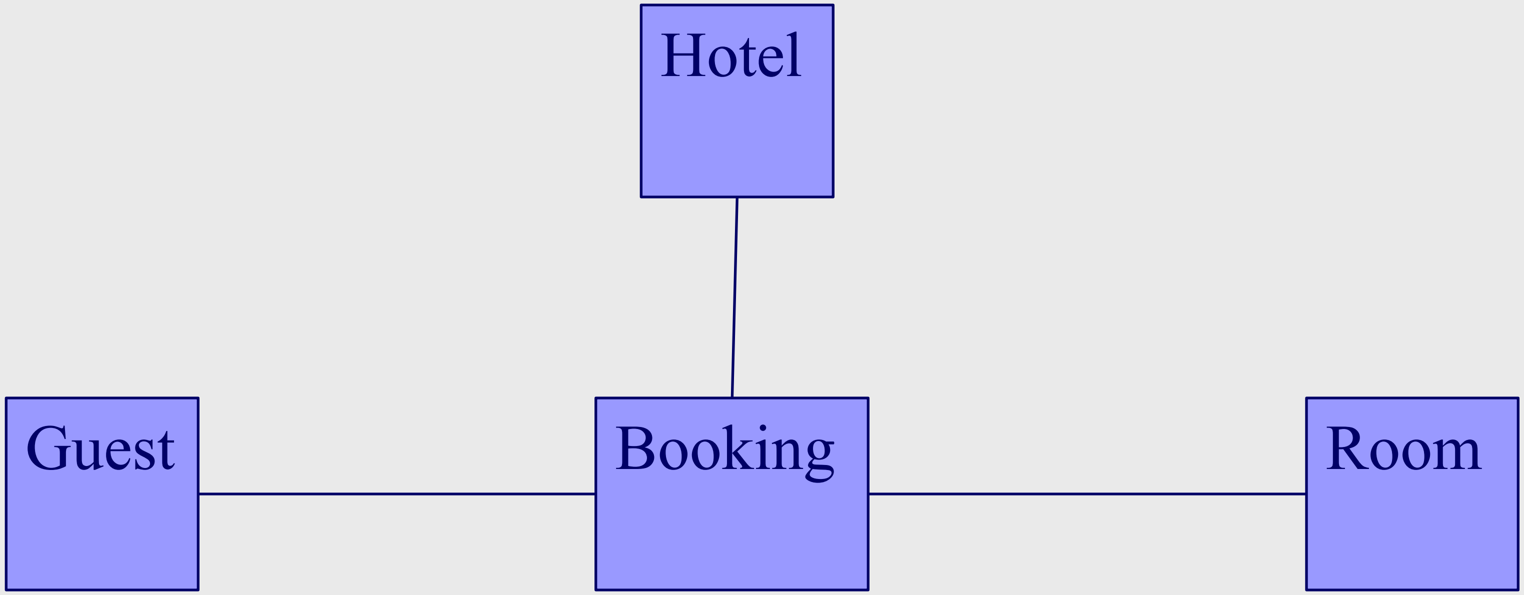


'Staff registers a client at a branch'

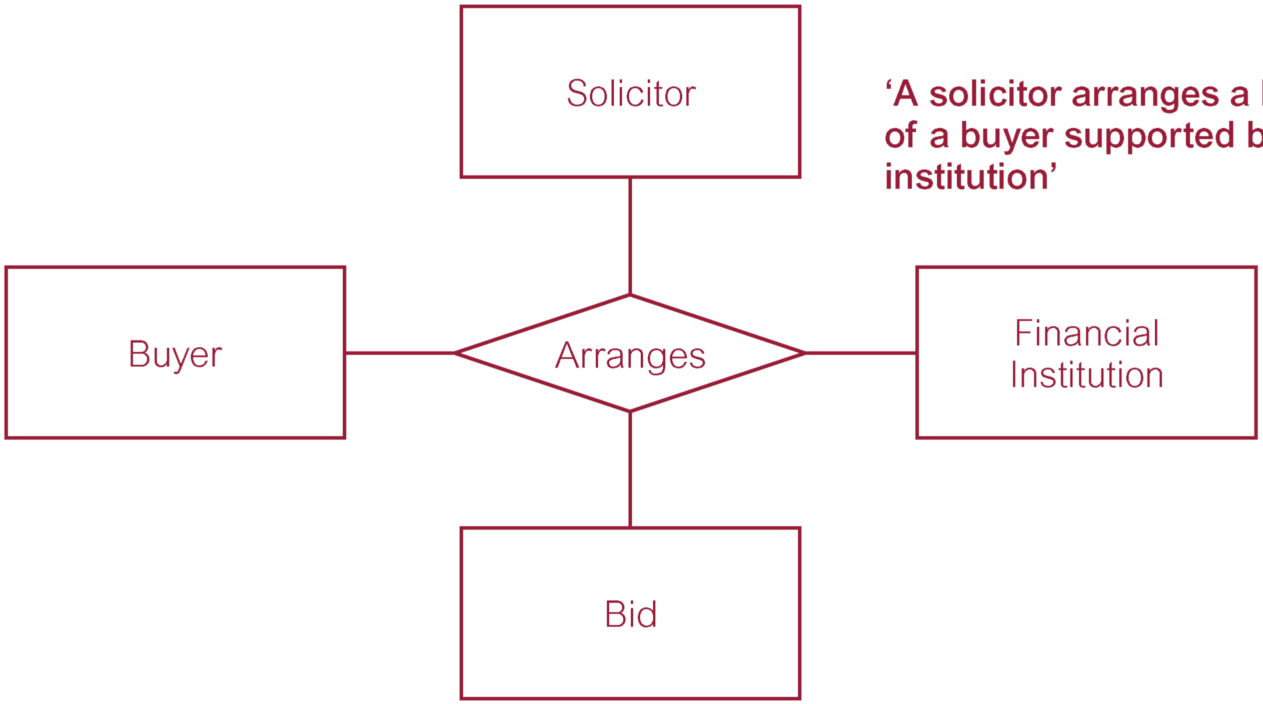
Another Ternary Relationship



Wrong way to show a ternary relationship



Quaternary relationship called *Arranges*



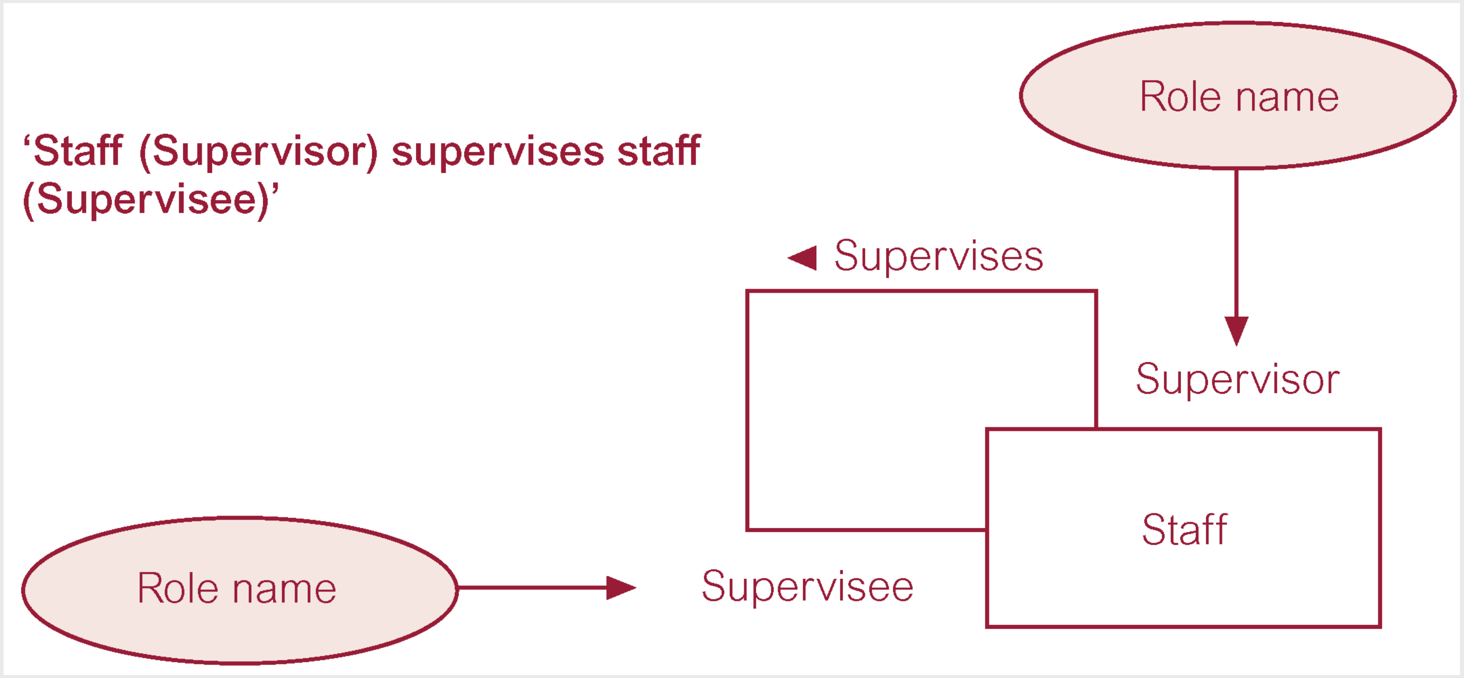
'A solicitor arranges a bid on behalf of a buyer supported by a financial institution'

Relationship Types

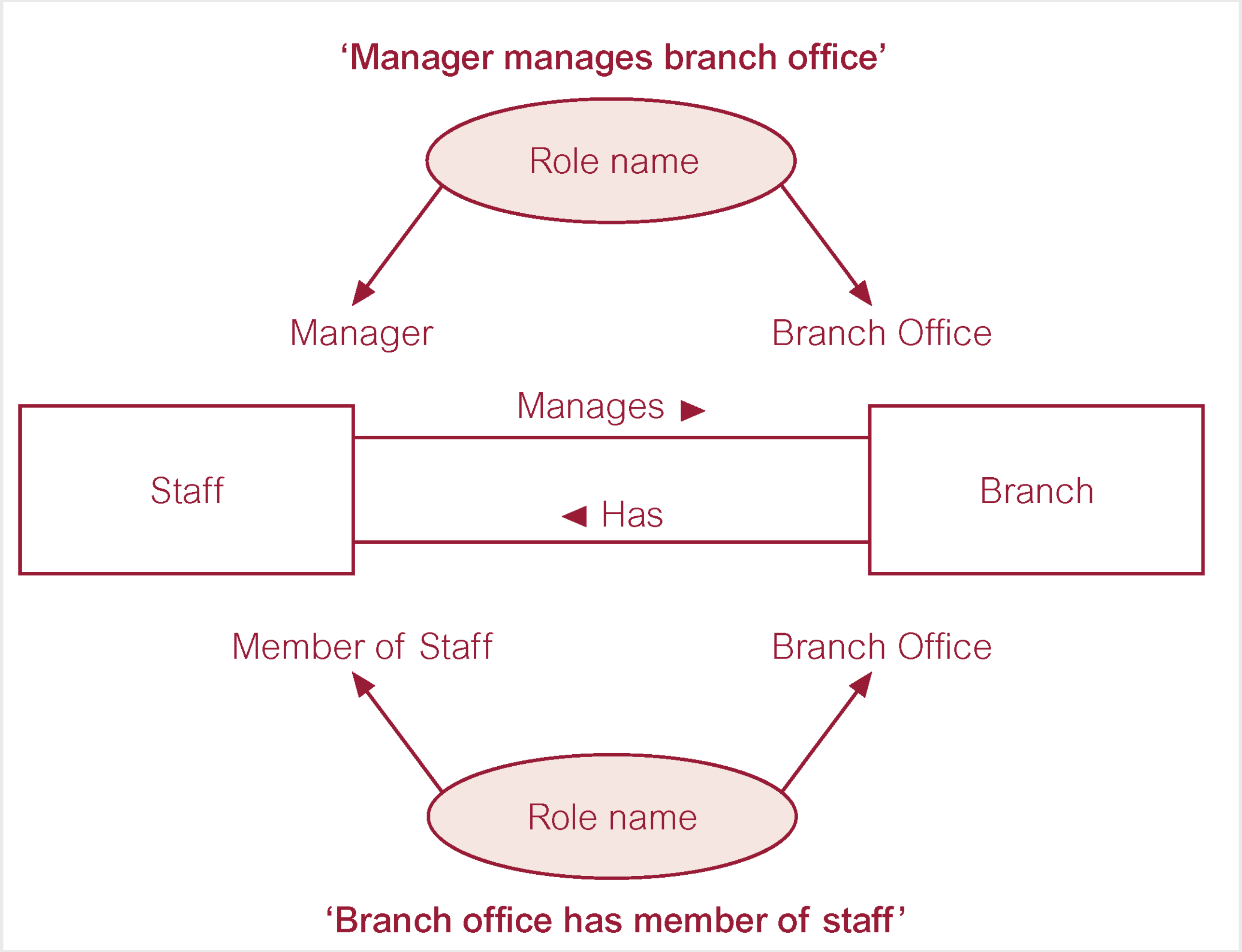
- Recursive Relationship
  - Relationship type where *same* entity type participates more than once in *different roles*.

- Relationships may be given role names to indicate purpose that each participating entity type plays in a relationship.

Recursive relationship called *Supervises* with role names



Entities associated through two distinct relationships with role names



Attributes

- A property of an entity or a relationship type.
- Types of attributes
  - » Descriptive attributes: provide descriptive information about the entity (unstable)
  - » Identifying attributes: provide information that uniquely identifies an entity (keys-stable)
  - » Foreign key attributes: Provides a link to another entity

Attributes

- Simple Attribute
  - Attribute composed of a single component with an independent existence.
- Composite Attribute
  - Attribute composed of multiple components, each with an independent existence.

Attributes

- Single-valued Attribute
  - Attribute that holds a single value for each occurrence of an entity type.
- Multi-valued Attribute
  - Attribute that holds multiple values for each occurrence of an entity type.
  - Typically are represented in a separate relation

Attributes

- **Derived Attribute**
  - **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 typically end up in views.

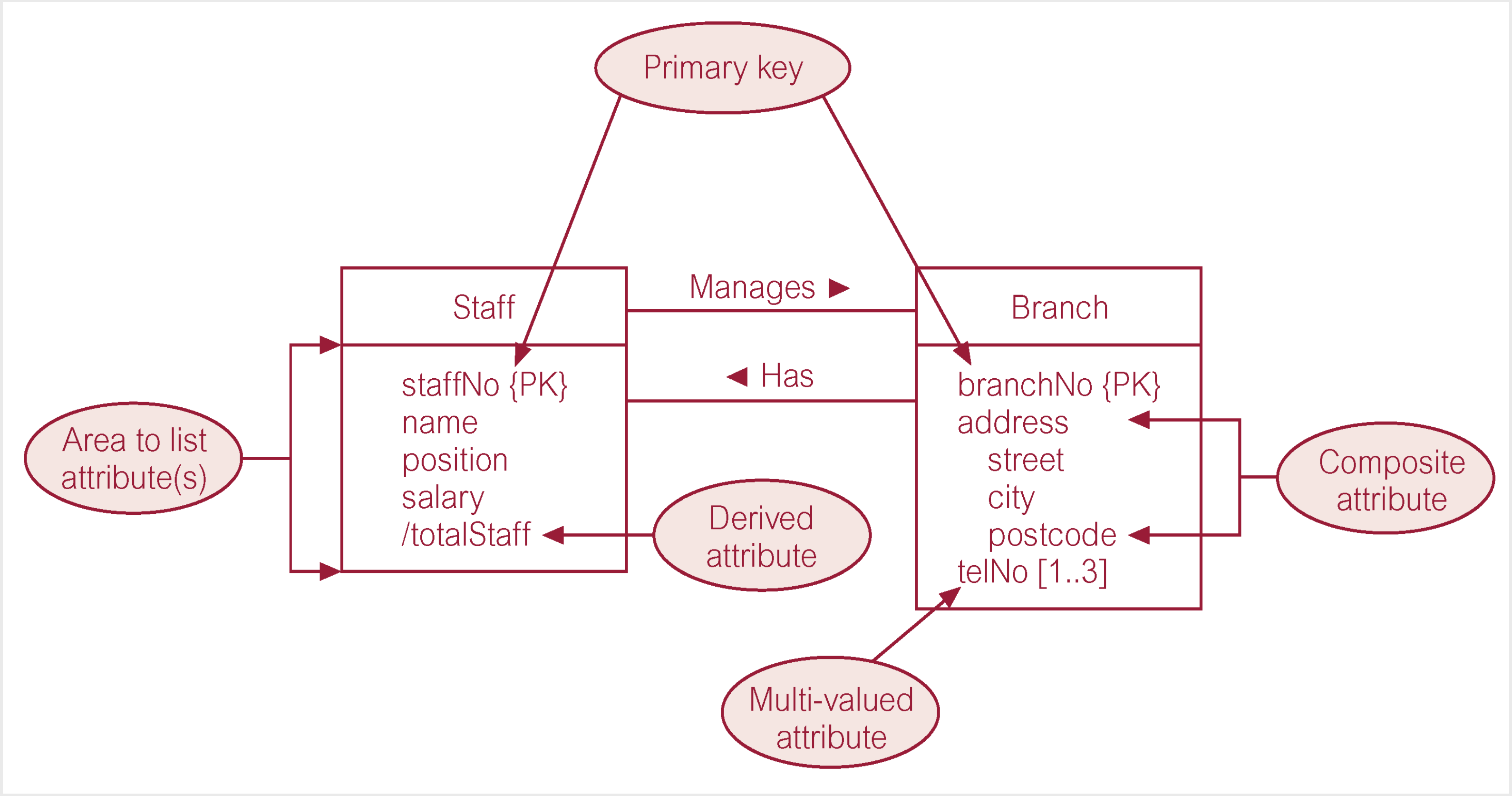
Derived attributes (cont)

- Types of derived attributes:
  1. Attributes computed from an aggregate function (e.g., total staff count)
  2. attribute computable from two columns in the same relation (e.g., duration of guest stay can be calculated from dateTo and dateFrom fields of booking relation)
  3. attribute computable from columns that belong to different relations (e.g., cost of guest stay is product of # days from booking and room cost from room)

Keys

- 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.

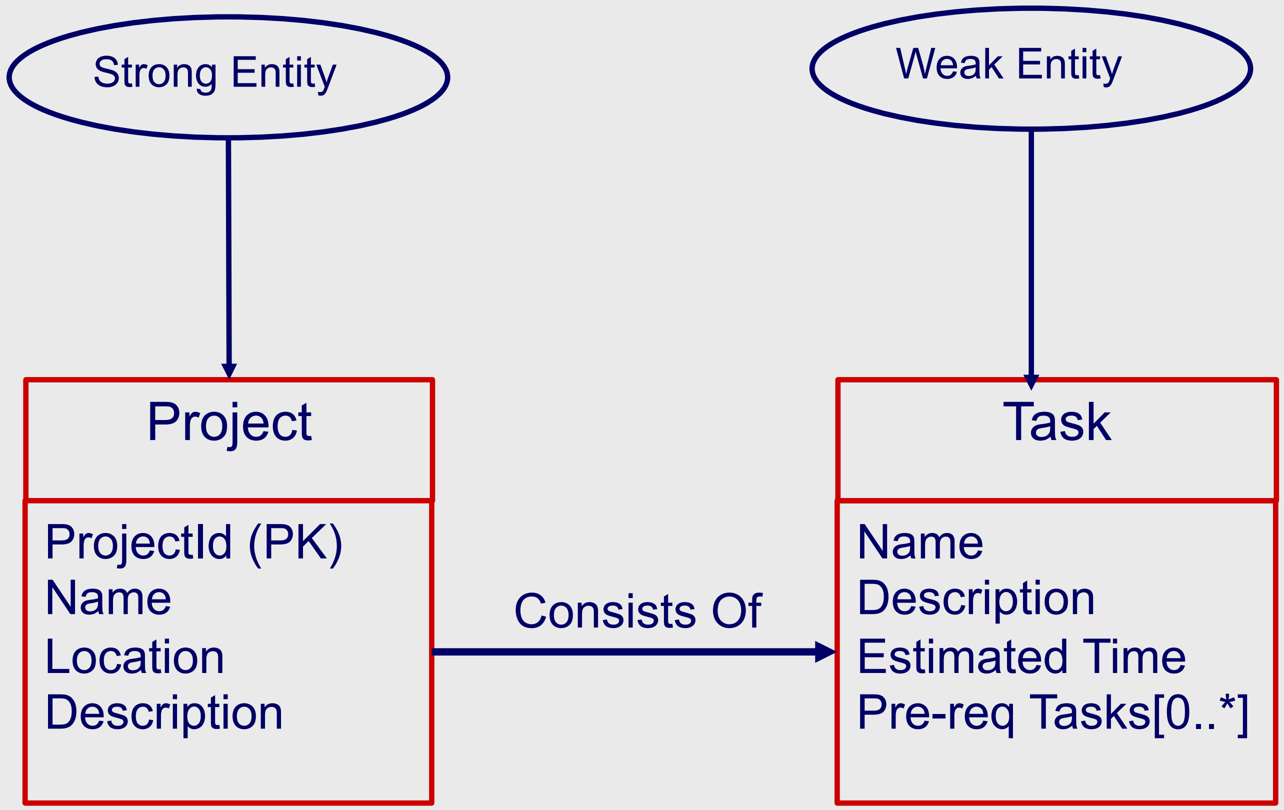
ER diagram of Staff and Branch entities and their attributes



Entity Type

- **Strong (master) Entity Type**
  - **Entity type that is *not* existence-dependent on some other entity type.**
- **Weak (dependent) Entity Type**
  - **Entity type that is existence-dependent on some other entity type.**

Strong entity type called Project and weak entity type called Task

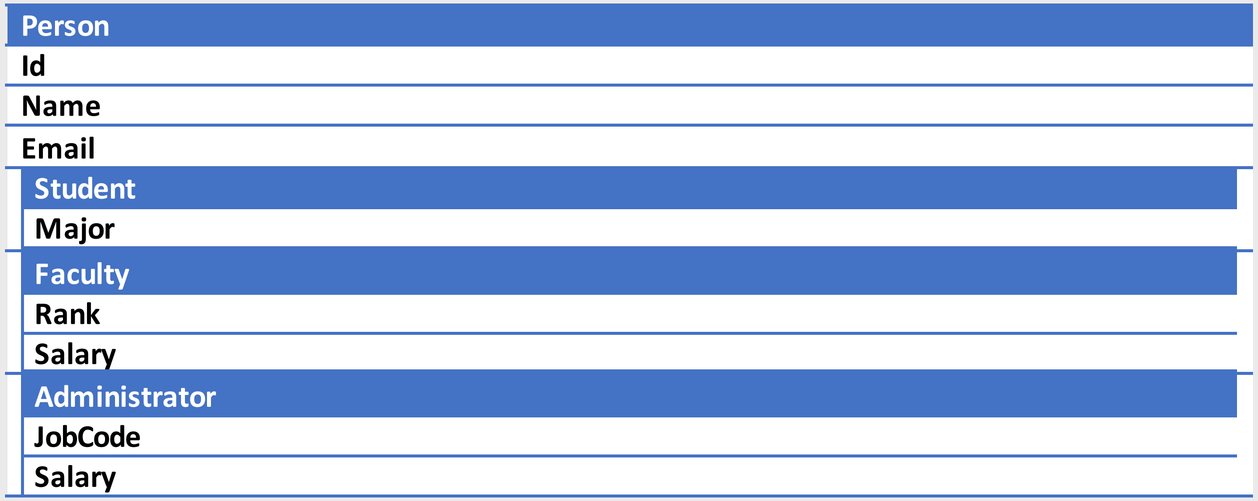


Supertype and Subtype Entities

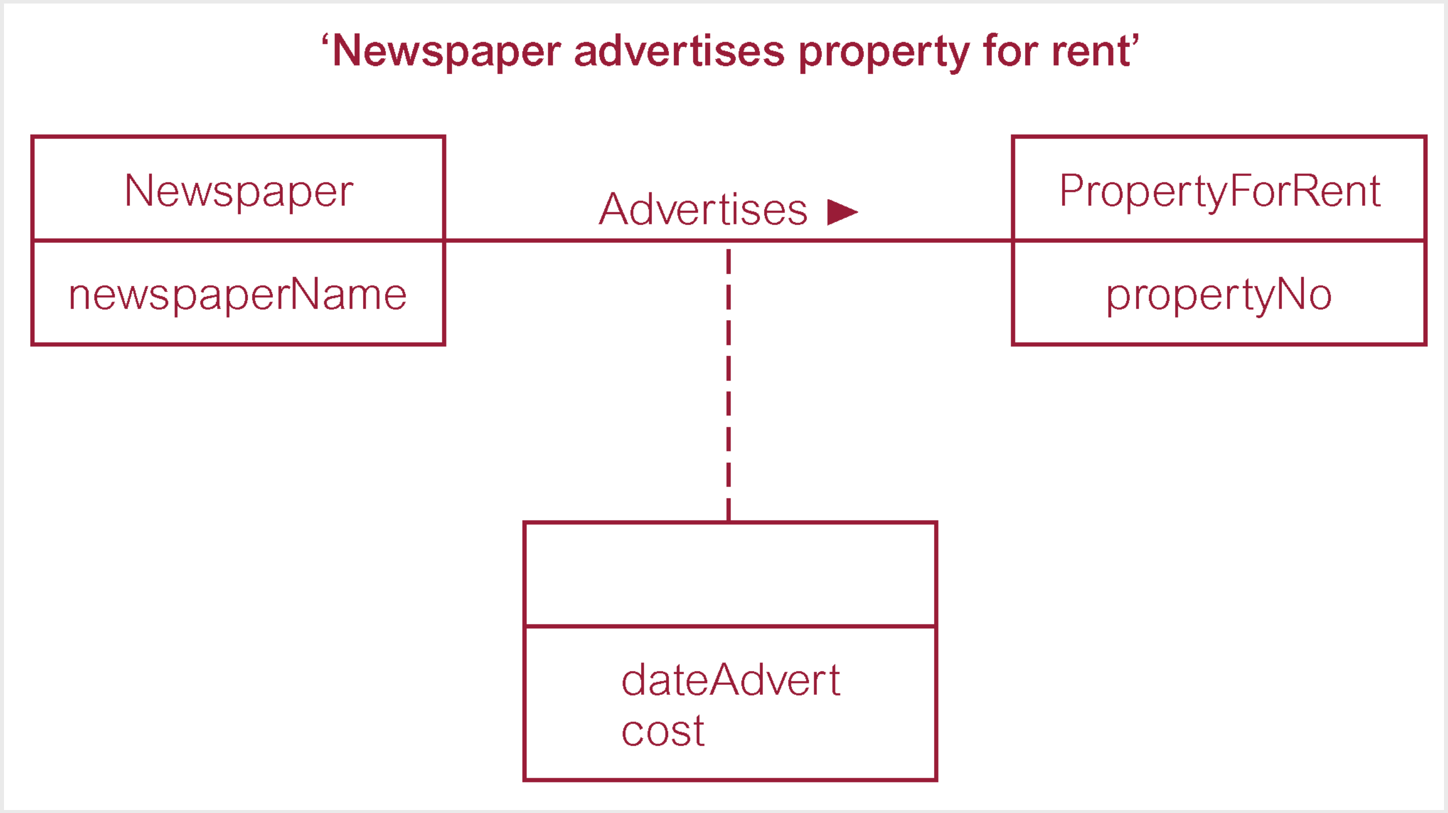
- Supertype Entity: A generic entity that provides a convenient grouping construct for multiple, related sub entities.
  - The sub entities share common attributes that are stored with the supertype entity
- Subtype Entity: An entity that is a subset of another entity

ER Representation

- Subtypes appear within supertype: Attributes are grouped with the appropriate entity



Relationship called *Advertises* with attributes



Structural Constraints

- **Main type of constraint on relationships is called *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.**
- **Represents policies (called *business rules*) established by user or company.**

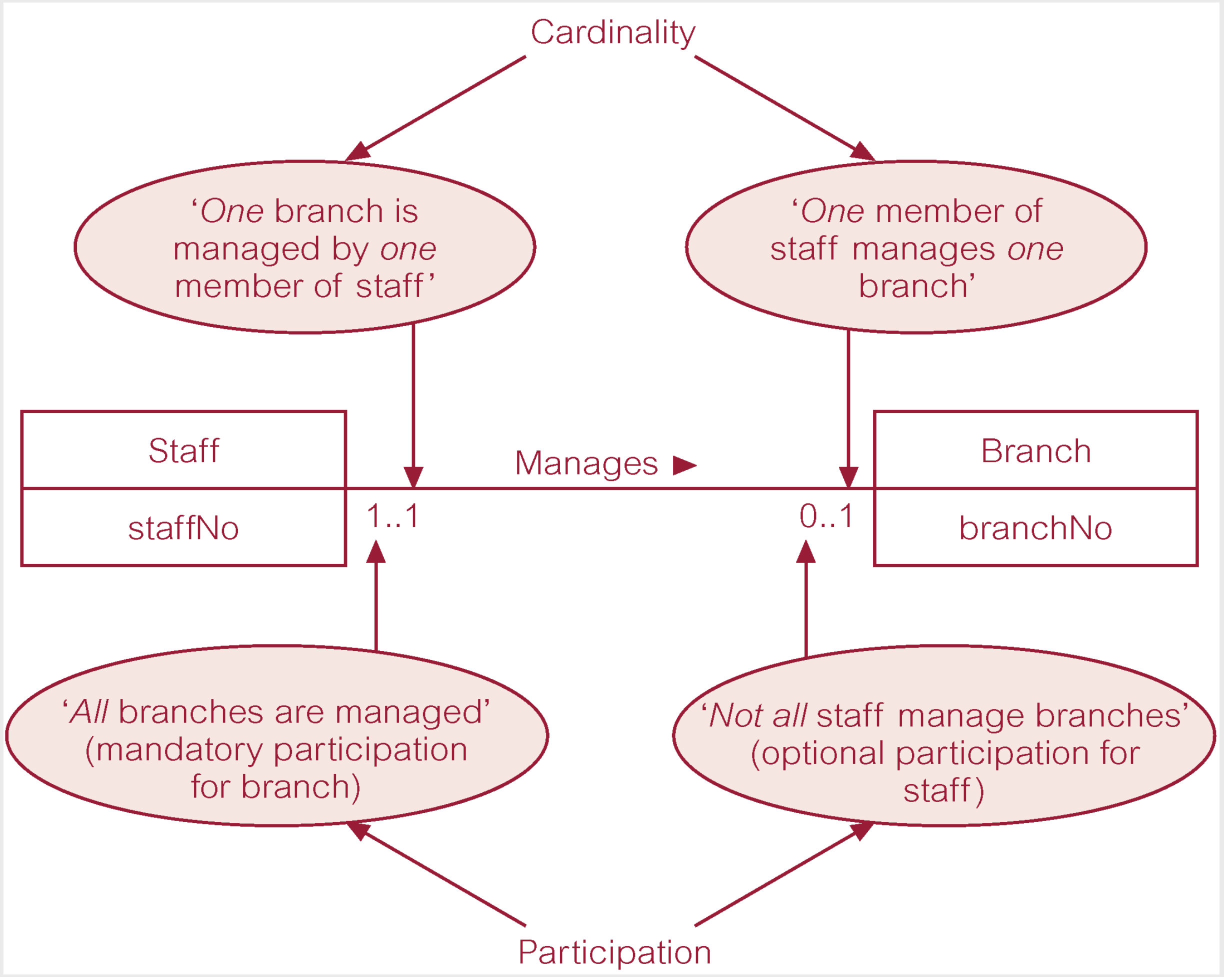
Structural Constraints

- **Multiplicity is made up of two types of restrictions on relationships: *cardinality* and *participation*.**

Structural Constraints

- Cardinality
  - Describes maximum number of possible relationship occurrences for an entity participating in a given relationship type.
- Participation
  - Determines the minimum number of entities that may participate in the relationship.
- Example: 0..1: means that a minimum of 0 and a maximum of 1 entity may occur in the relationship

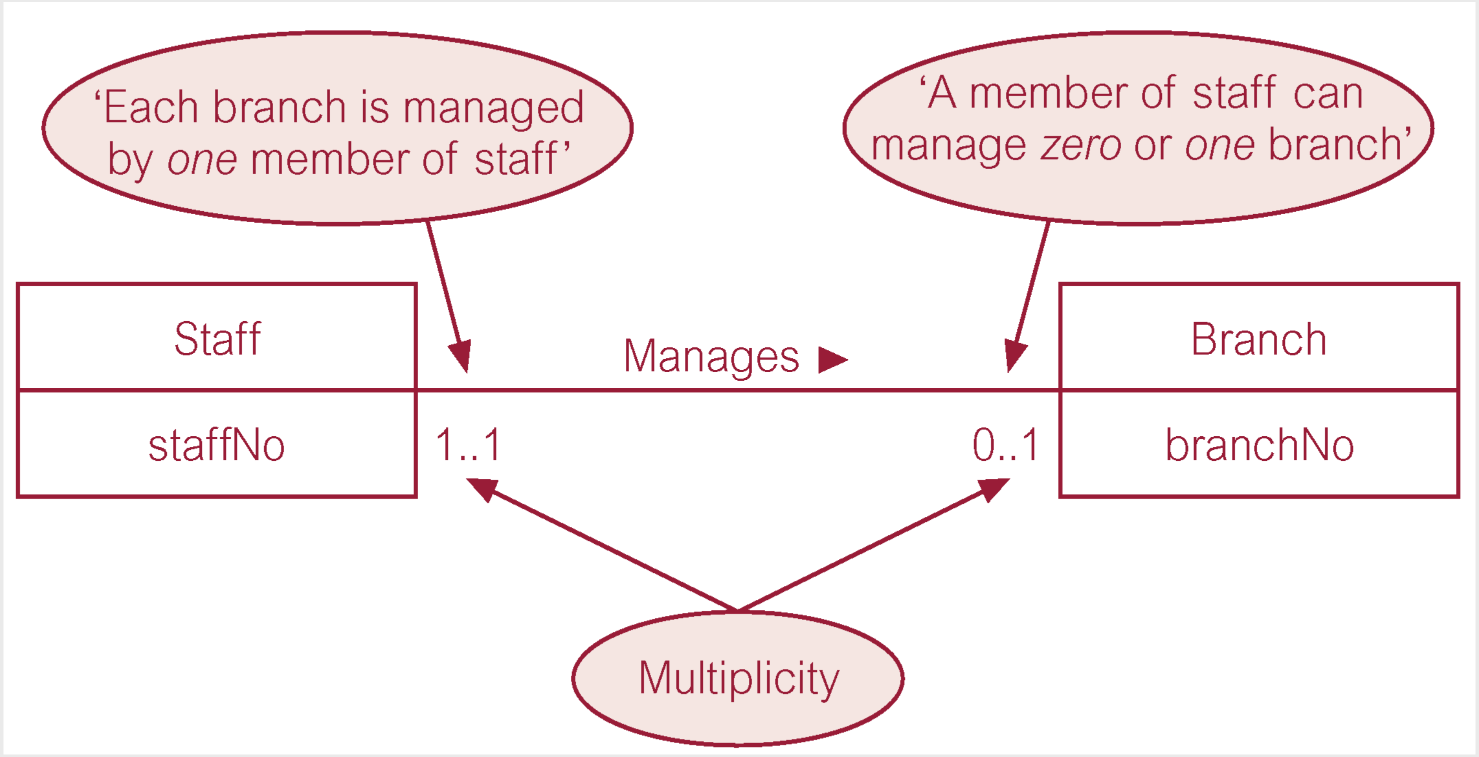
Multiplicity as cardinality and participation constraints


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

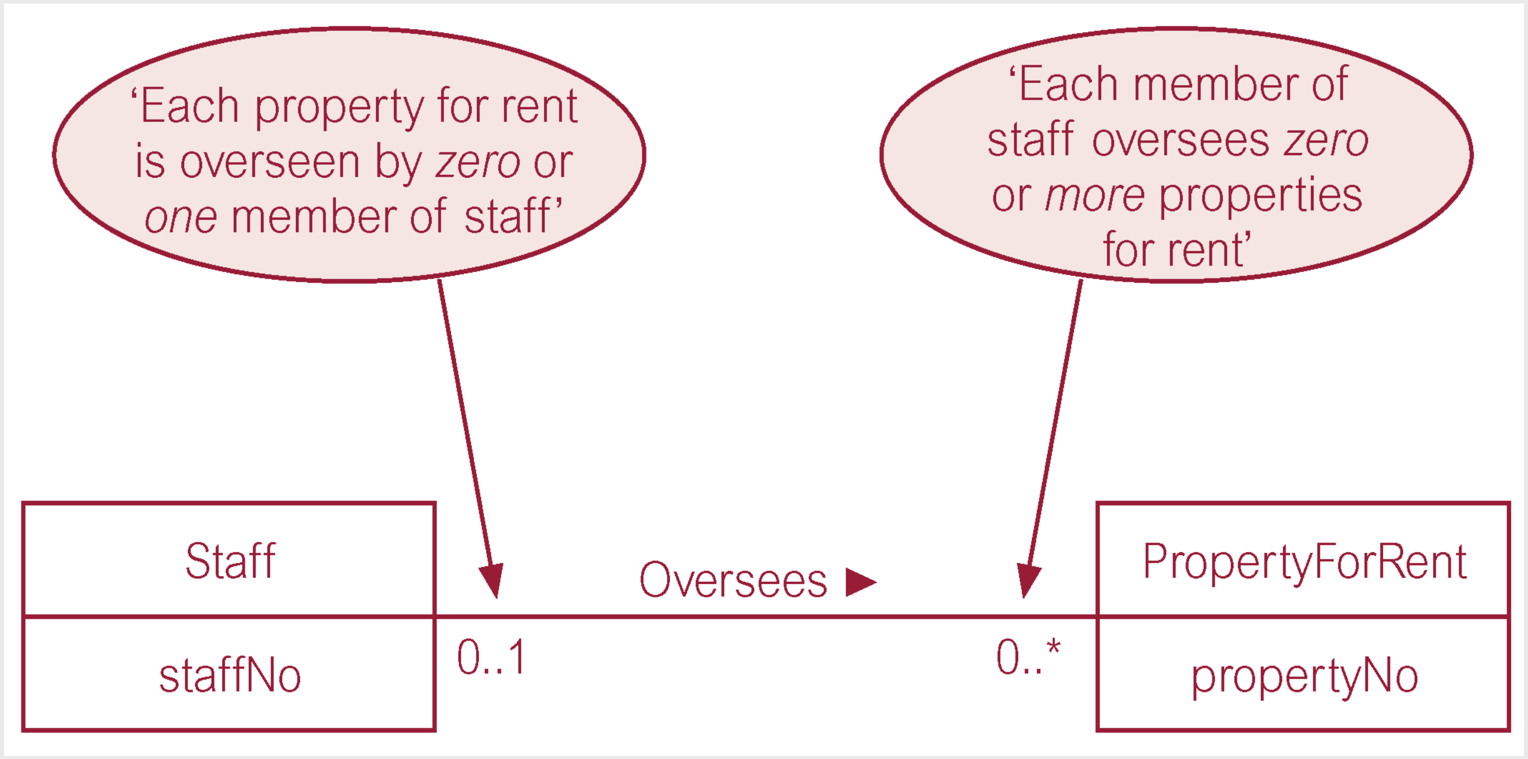
Structural Constraints

- 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 (*:*)

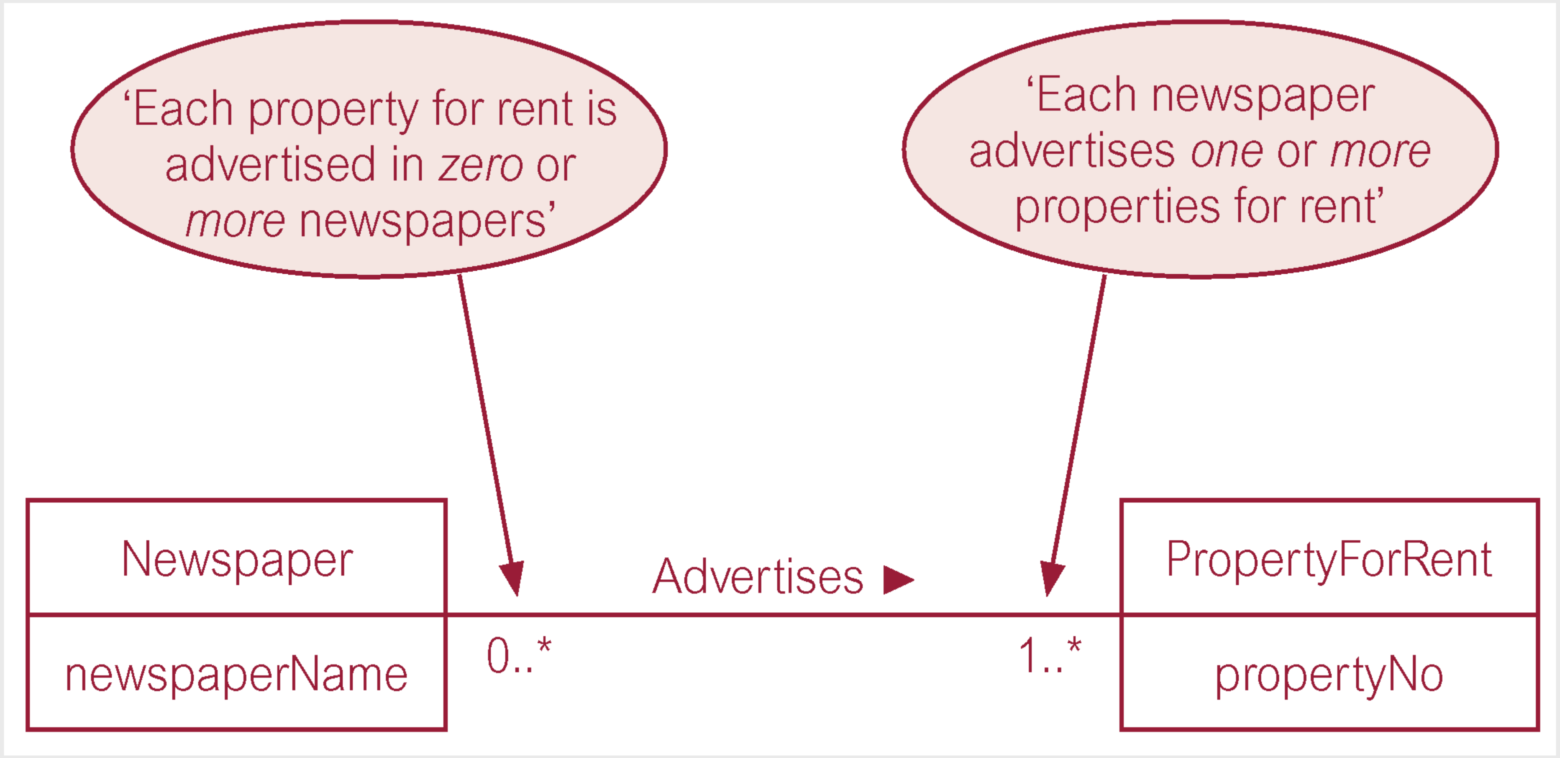
Multiplicity of Staff *Manages* Branch (1:1) relationship


'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

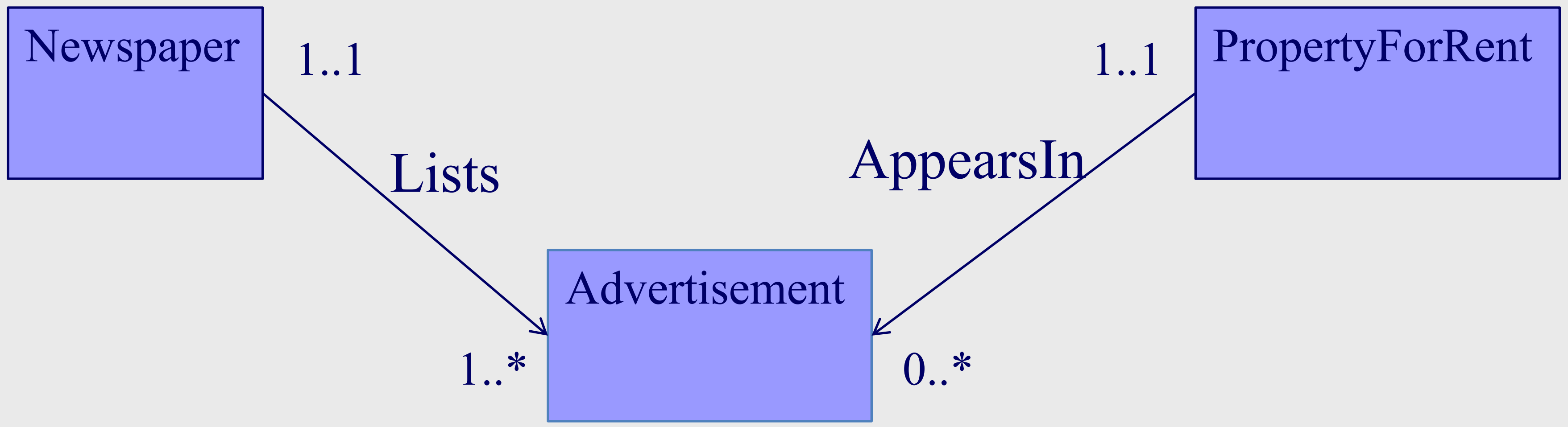
Multiplicity of Staff *Oversees* PropertyForRent (1:*) relationship type


'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

Multiplicity of 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

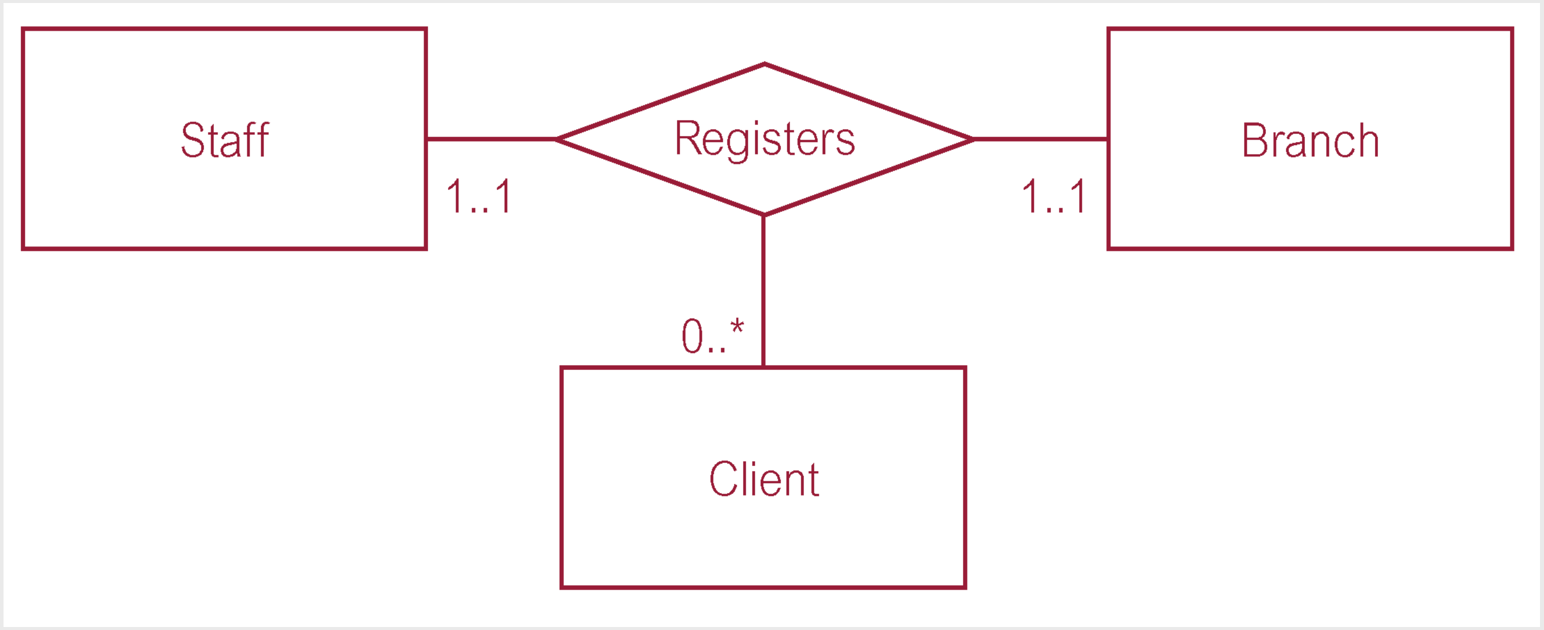
The Wrong Way to Show a *:* Relationship



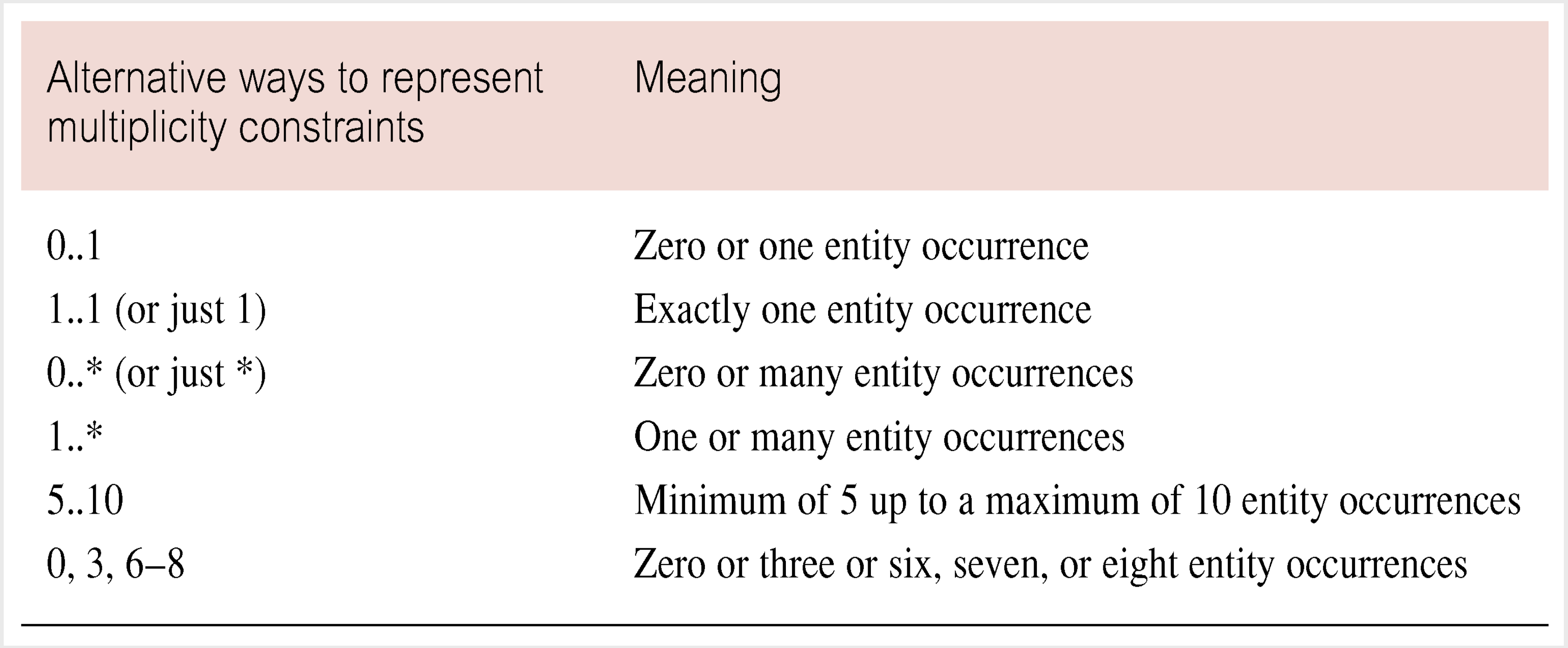
Structural Constraints

- **Multiplicity for Complex Relationships**
  - **Number (or range) of possible occurrences of an entity type in an *n*-ary relationship when other (*n*-1) values are fixed.**
  - **May help to write the relationship as a function.**
    - **Example: There is a relationship D among 3 entities A, B, C.**
      - **If |D(A,B)| = 0..1, then you put 0..1 next to C**

Multiplicity of ternary *Registers* relationship



Summary of multiplicity constraints

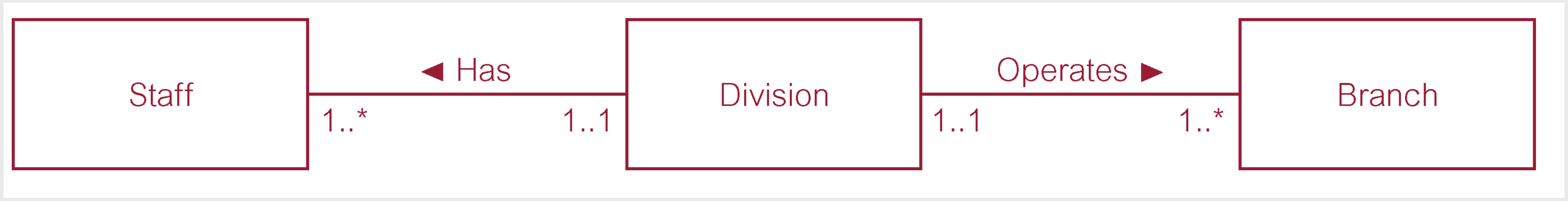


Alternative ways to represent multiplicity constraints	Meaning
0..1	Zero or one entity occurrence
1..1 (or just 1)	Exactly one entity occurrence
0..* (or just *)	Zero or many entity occurrences
1..*	One or many entity occurrences
5..10	Minimum of 5 up to a maximum of 10 entity occurrences
0, 3, 6–8	Zero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

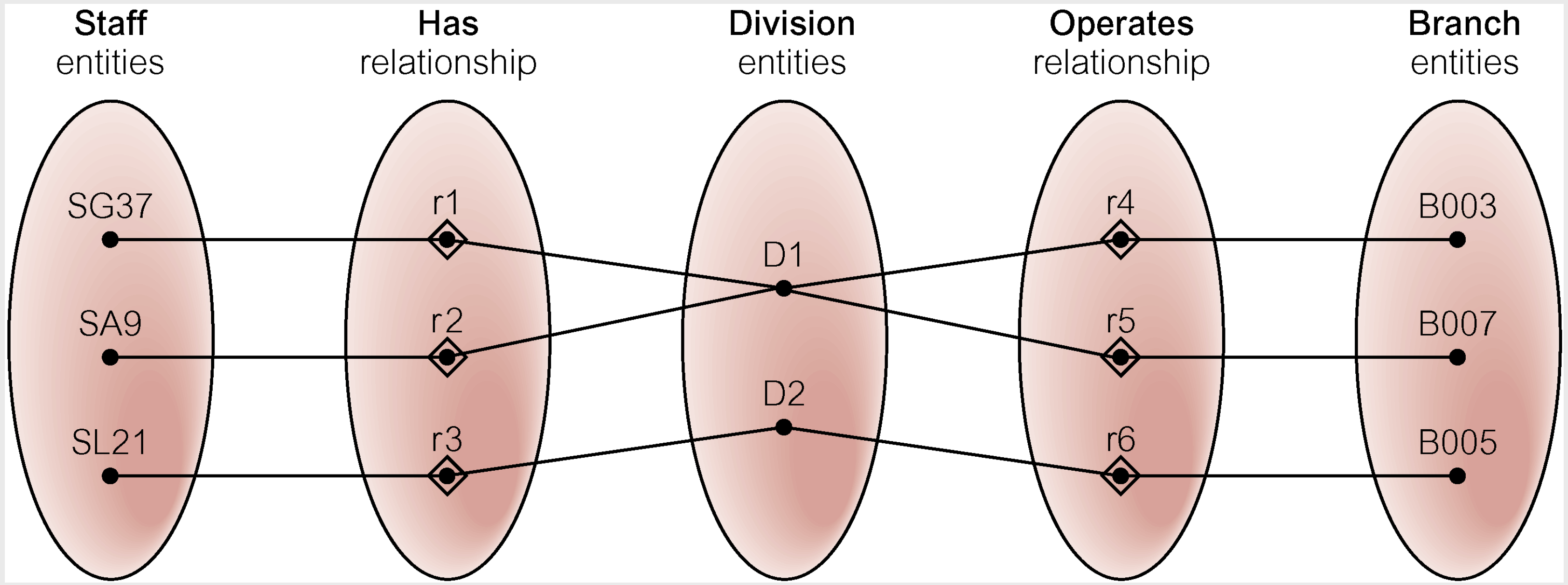
Problems with ER Models

- Problems may arise when designing a conceptual data model called *connection traps*.
- Often due to a misinterpretation of the meaning of certain relationships.
- Two main types of connection traps are called *fan traps* and *chasm traps*.

An Example of a Fan Trap

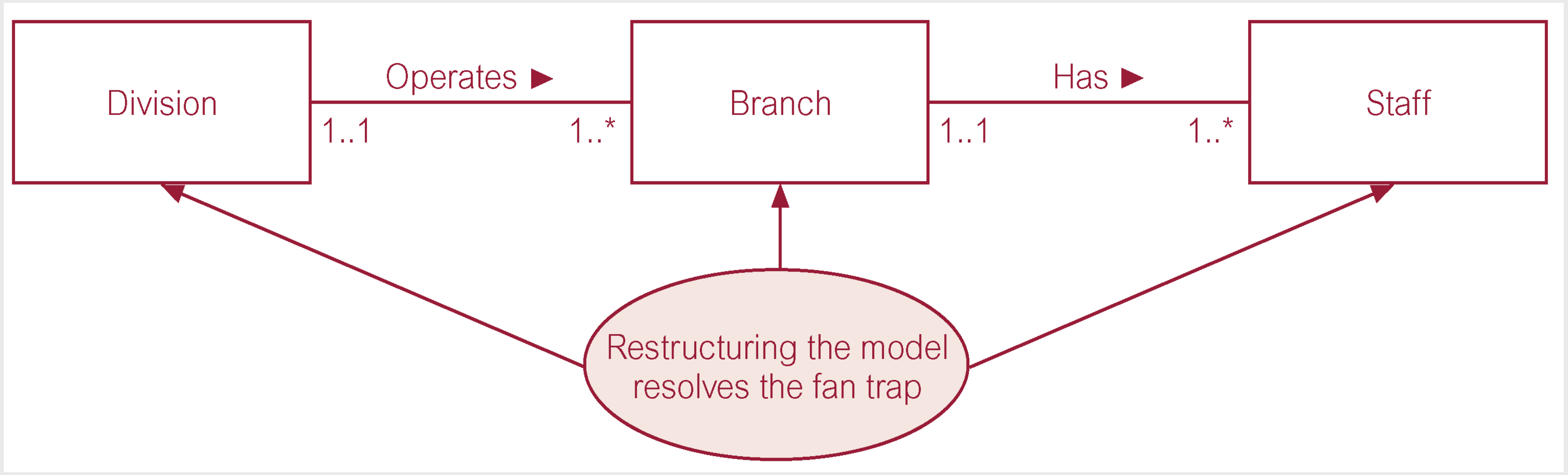


Semantic Net of ER Model with Fan Trap

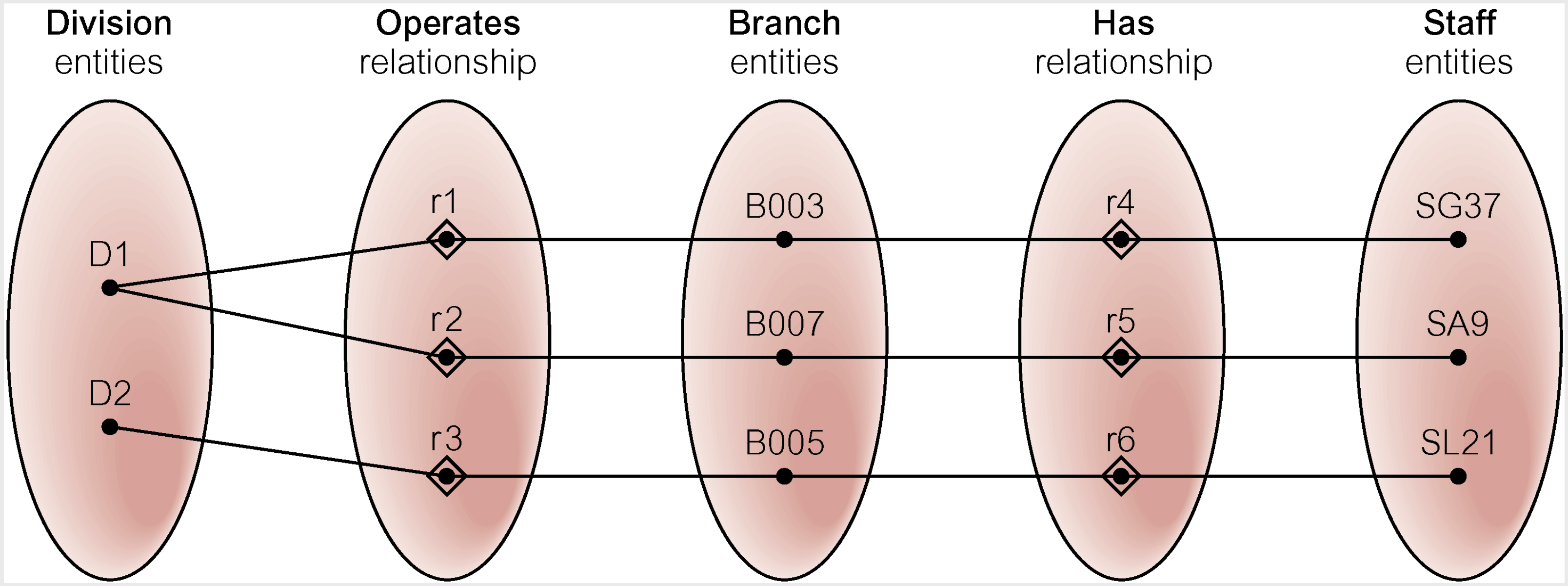


- **At which branch office does staff number SG37 work?**

Restructuring ER model to remove Fan Trap



Semantic Net of Restructured ER Model with Fan Trap Removed



- **SG37 works at branch B003.**

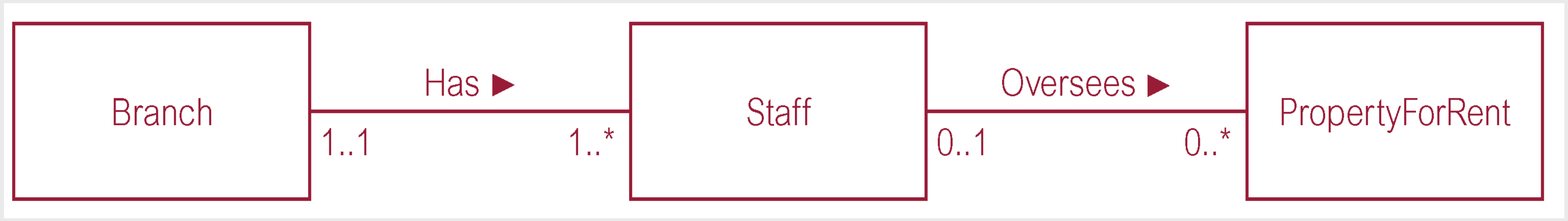
Fan Trap Summary

- Where a model represents a relationship between entity types, but pathway creates a “choke point” that eliminates information
- Fan traps usually occur when there are multiple 1..* relationships leaving an entity and at least one of the 1..* relationships involves a transitive relationship
- The usual solution is to redirect the pathways so that the diagram shows only direct relationships rather than transitive relationships

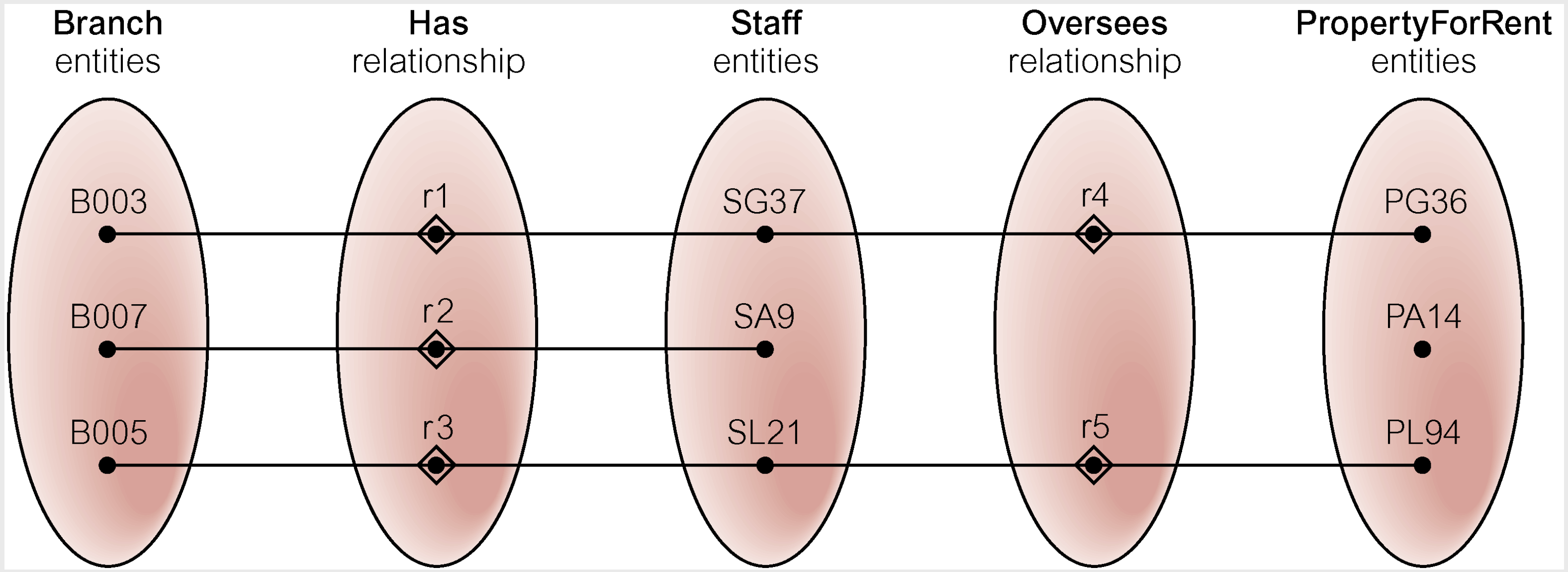
Chasm Trap

- **Where a model suggests the existence of a relationship between entity types, but pathway does not exist between certain entity occurrences.**
  - **There appears to be an unbroken path between two entities, but somewhere on that path there is a 0 participation rate**

An Example of a Chasm Trap

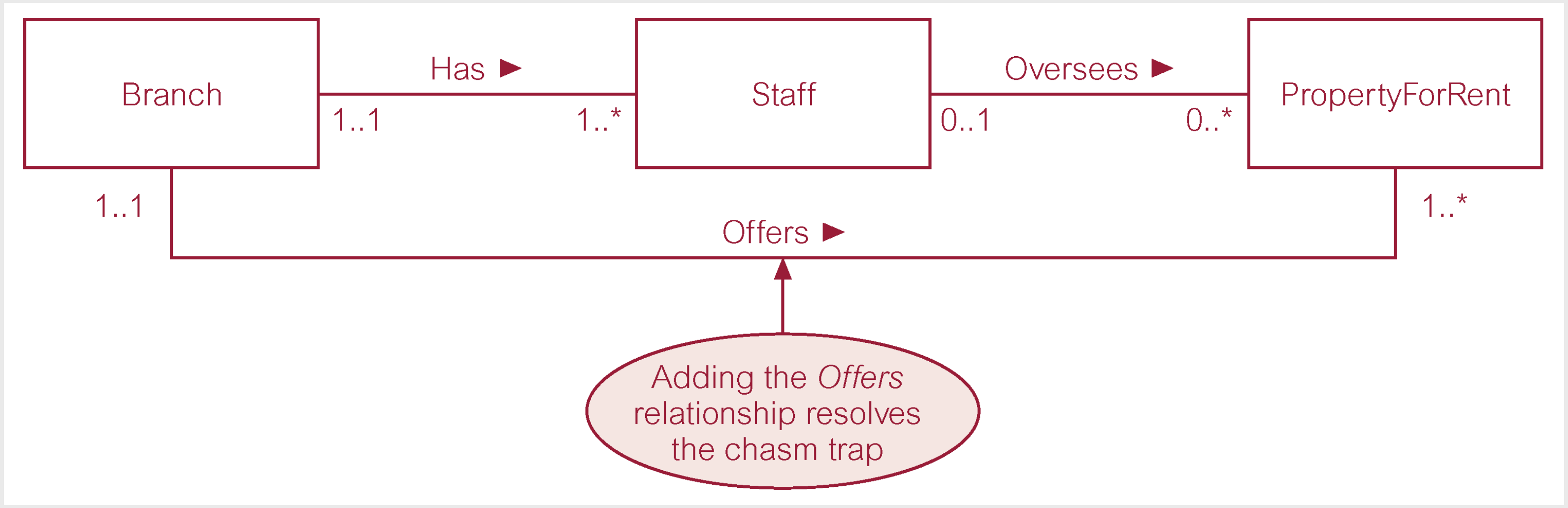


Semantic Net of ER Model with Chasm Trap

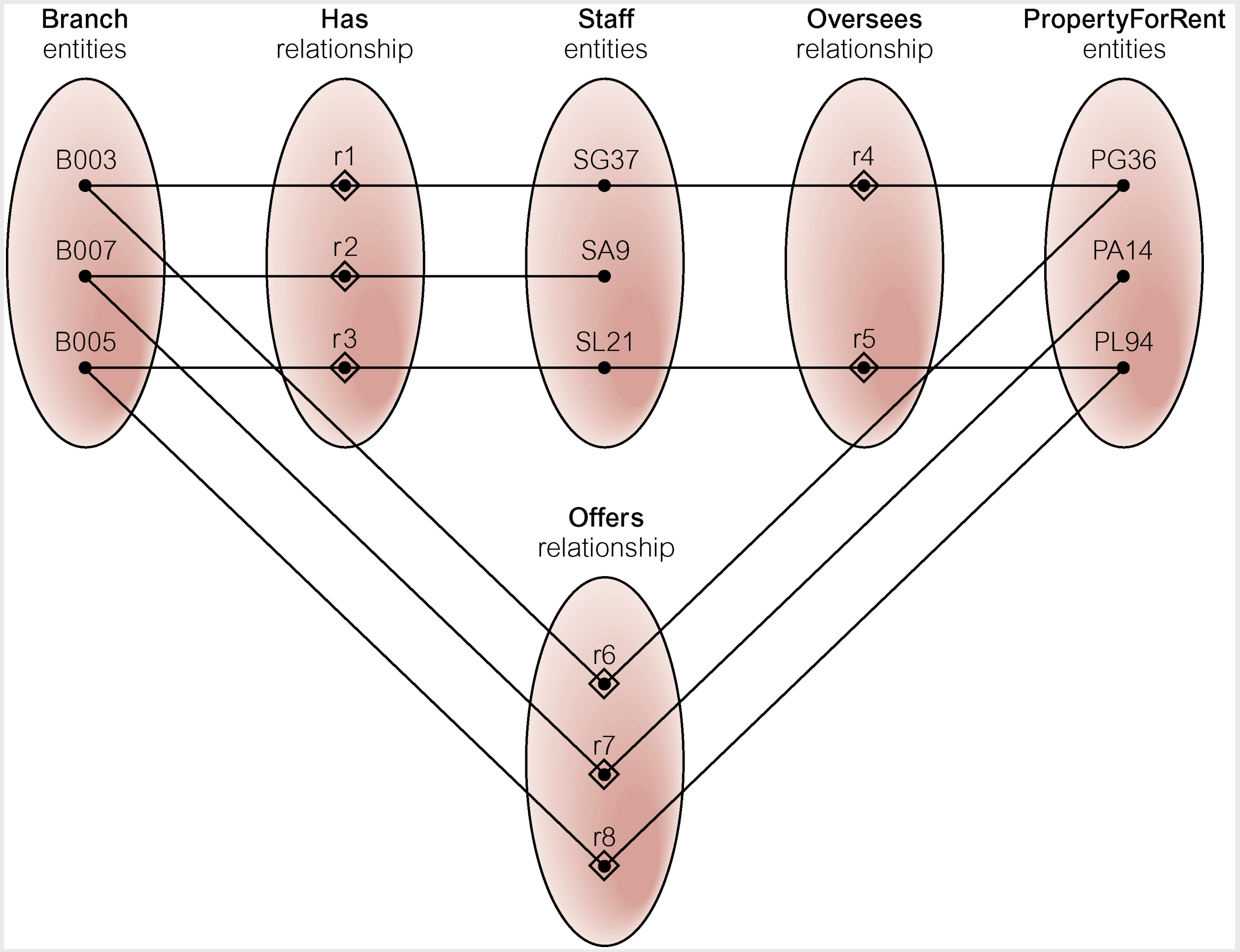


- **At which branch office is property PA14 available?**

ER Model restructured to remove Chasm Trap



Semantic Net of Restructured ER Model with Chasm Trap Removed



Chasm Trap Summary

- Occurs when an ER diagram suggests the existence of a transitive relationship between entity types, but the pathway does not always exist because somewhere on that path there is a 0 participation rate
- Ways to fix the trap
  - Add a direct relationship between the two entities that are disconnected by the trap—problem is that this solution adds redundant data to the database
  - Convince the customer to allow a default value so that the participation rate can be changed to 1