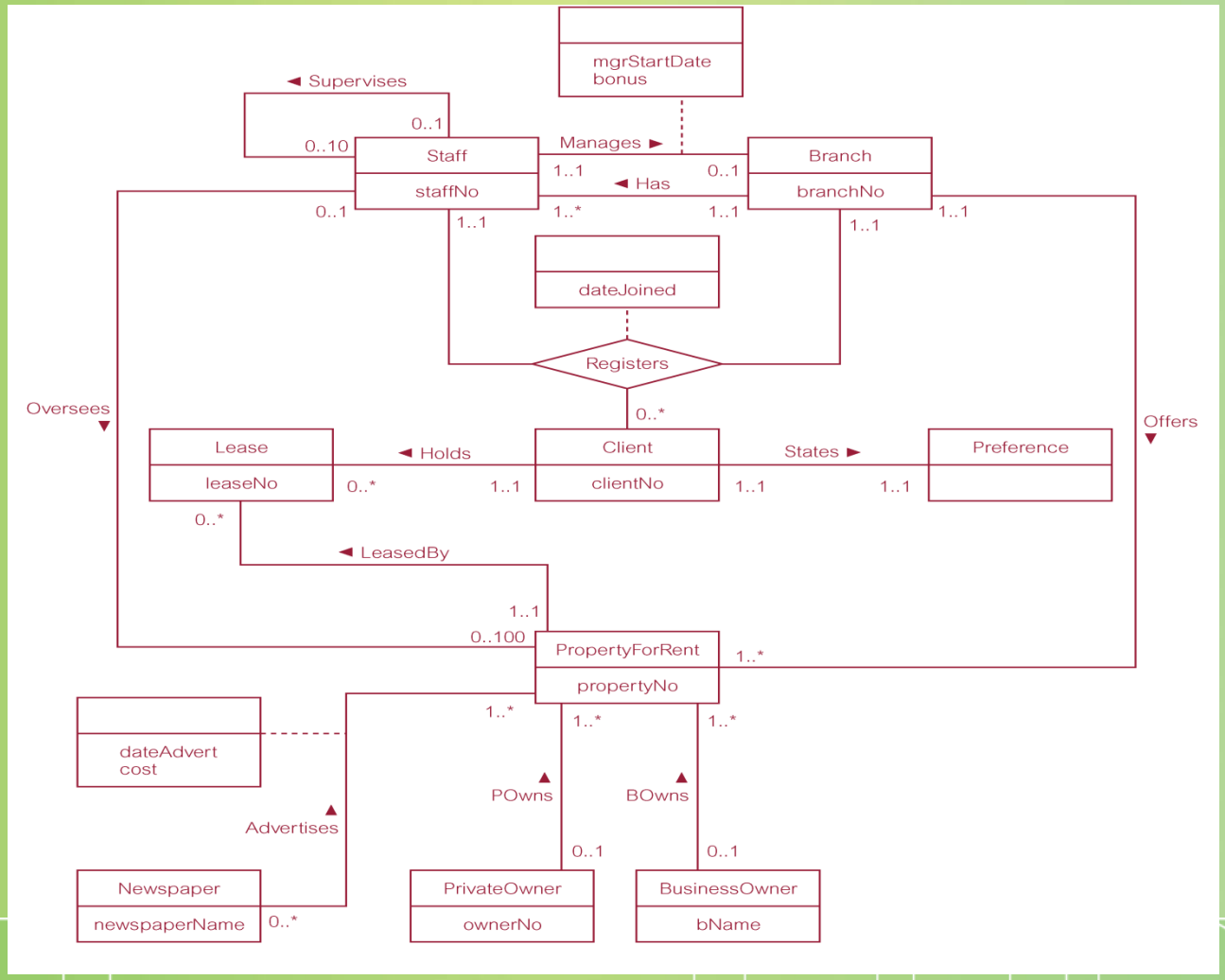
Chapter 12

Entity-Relationship Modeling

Chapter 12 - Objectives

- How to use Entity–Relationship (ER) modeling in database design.
- 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.
- How to build an ER model from a requirements specification.

ER diagram of Branch user views of *DreamHome*



Concepts of the ER Model

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

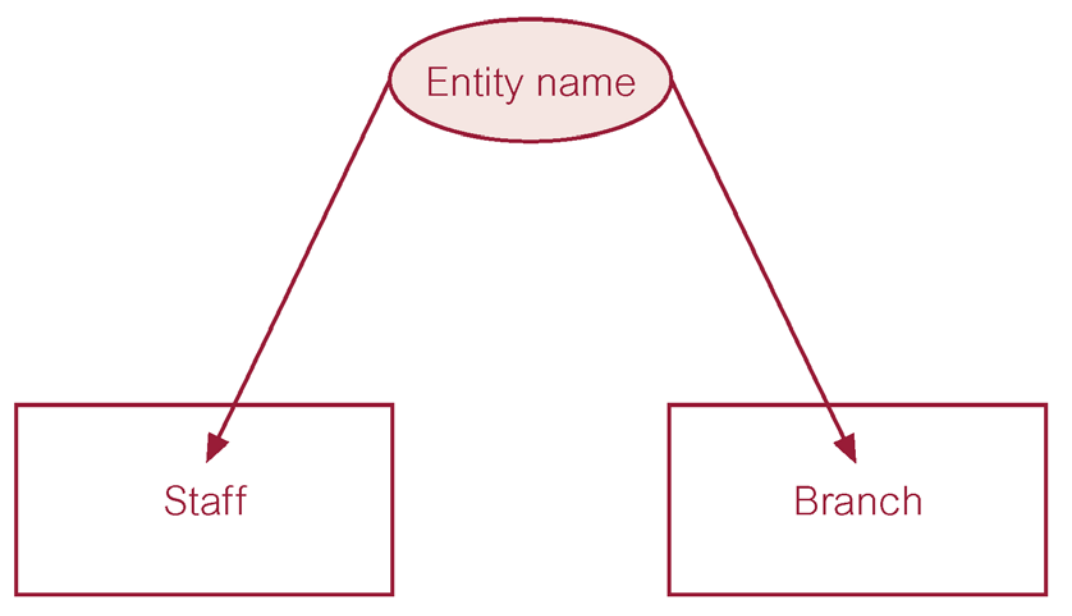
Entity Type

- **Entity type**
  - **Group of objects with same properties, identified by enterprise as having an independent existence.**
- **Entity occurrence**
  - **Uniquely identifiable object of an entity type.**

Examples of Entity Types

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

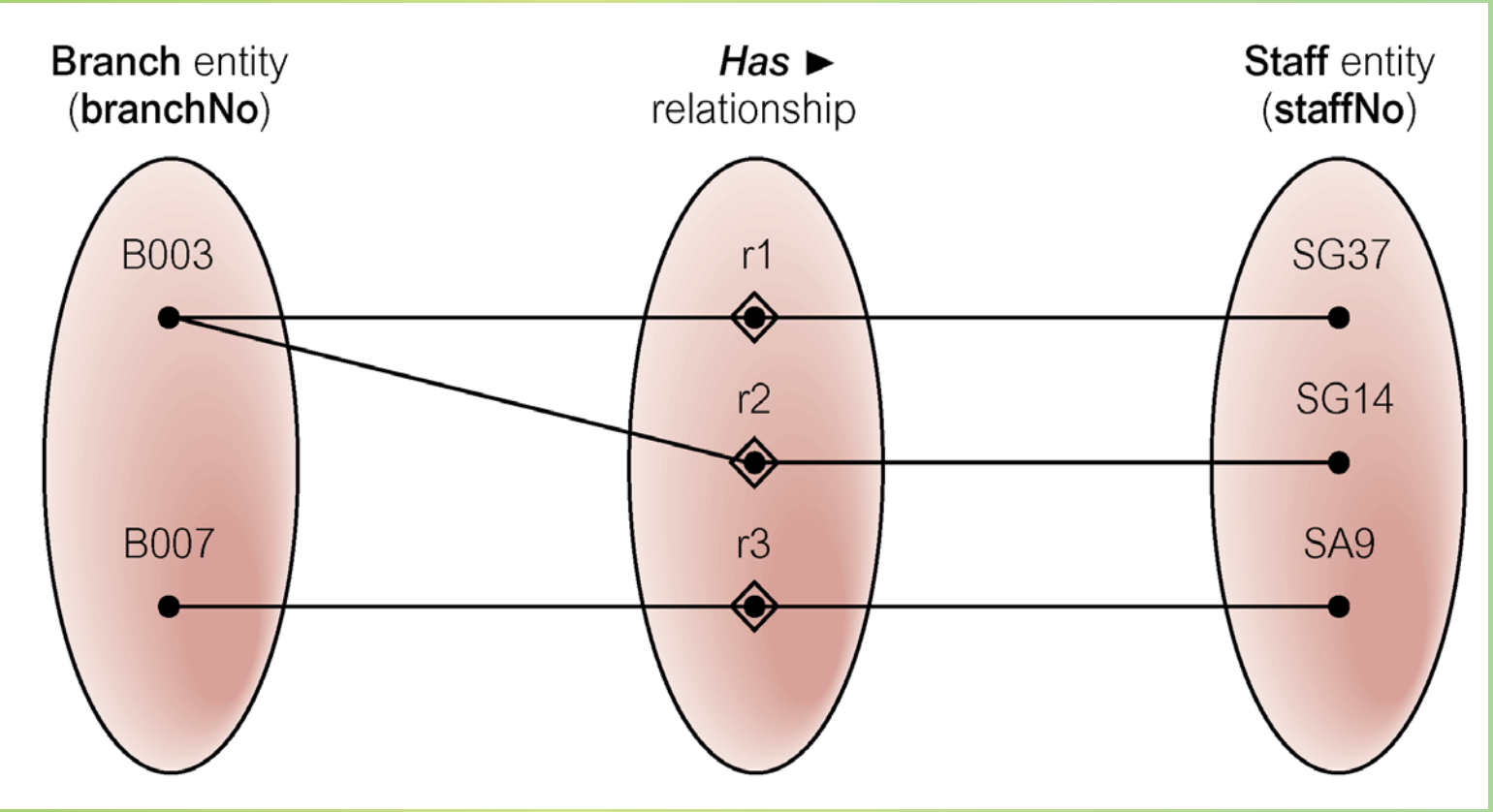
ER diagram of Staff and Branch entity types



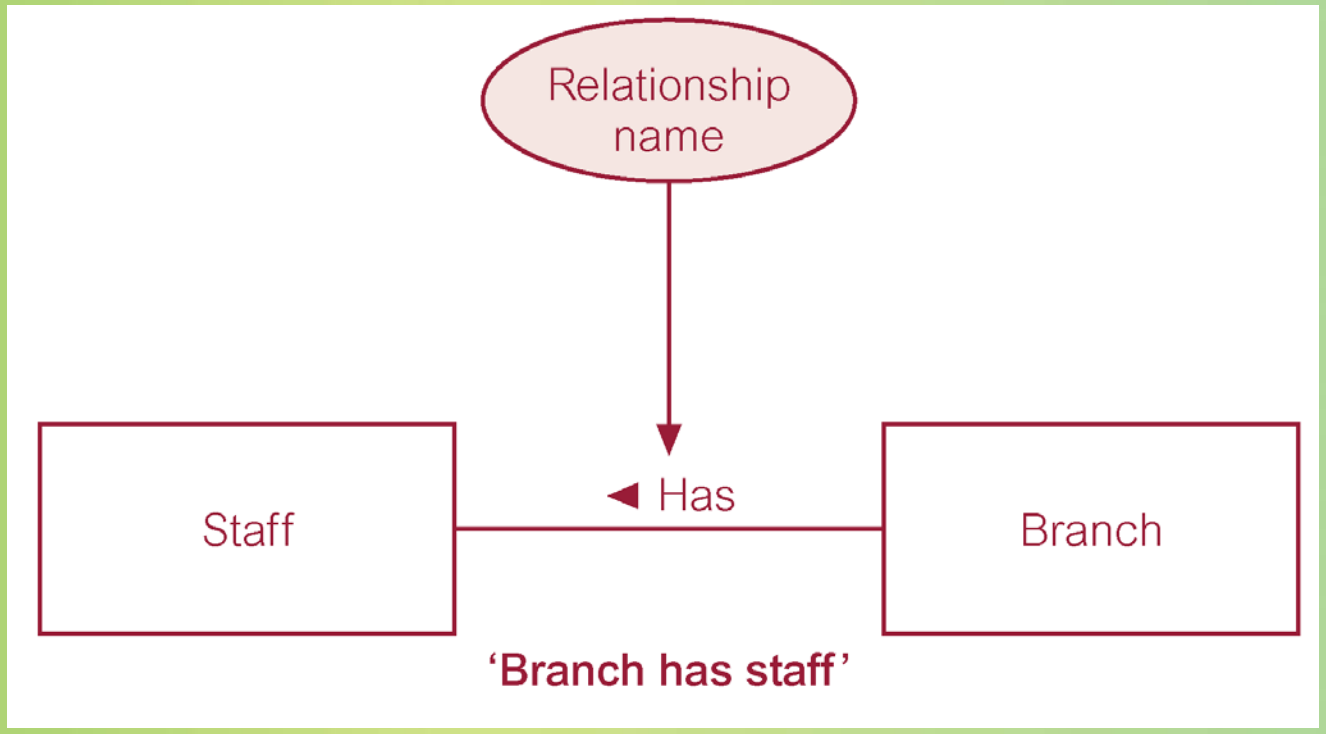
Relationship Types

- **Relationship type**
  - **Set of meaningful associations among entity types.**
- **Relationship occurrence**
  - **Uniquely identifiable association, which includes one occurrence from each participating entity type.**

Semantic net of *Has* relationship type



ER diagram of Branch *Has* Staff relationship

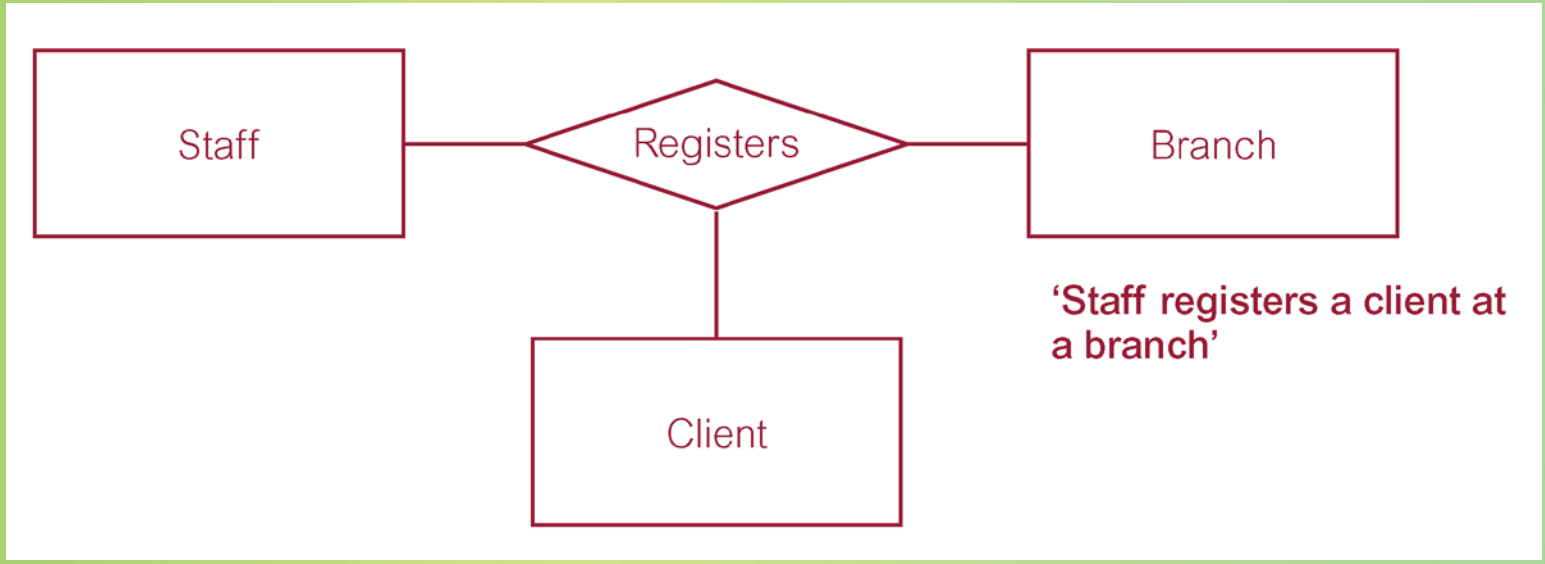


Relationship Types

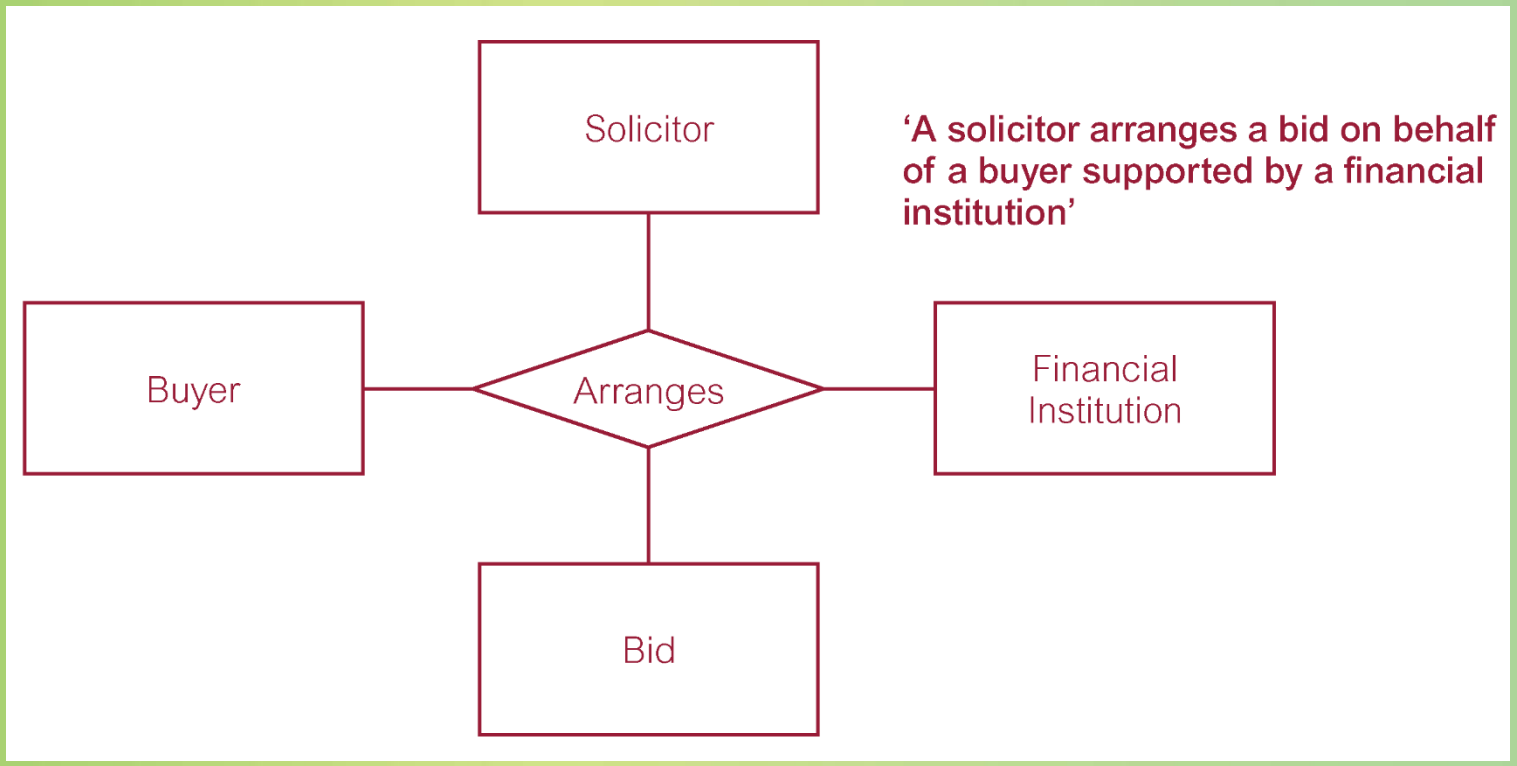
- **Degree of a Relationship**
  - **Number of participating entities in relationship.**
- **Relationship of degree :**
  - **two is binary**
  - **three is ternary**
  - **four is quaternary.**

Binary relationship called *POwns*

Ternary relationship called *Registers*



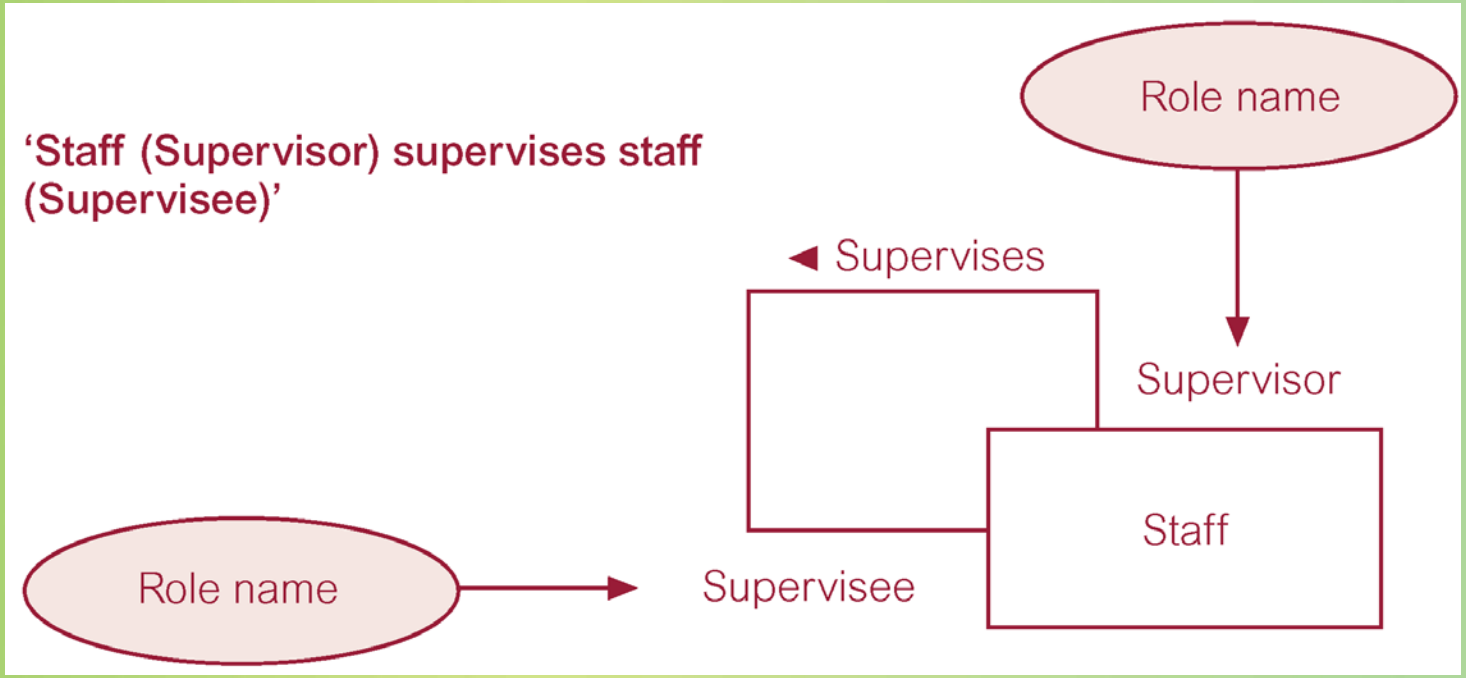
Quaternary relationship called *Arranges*



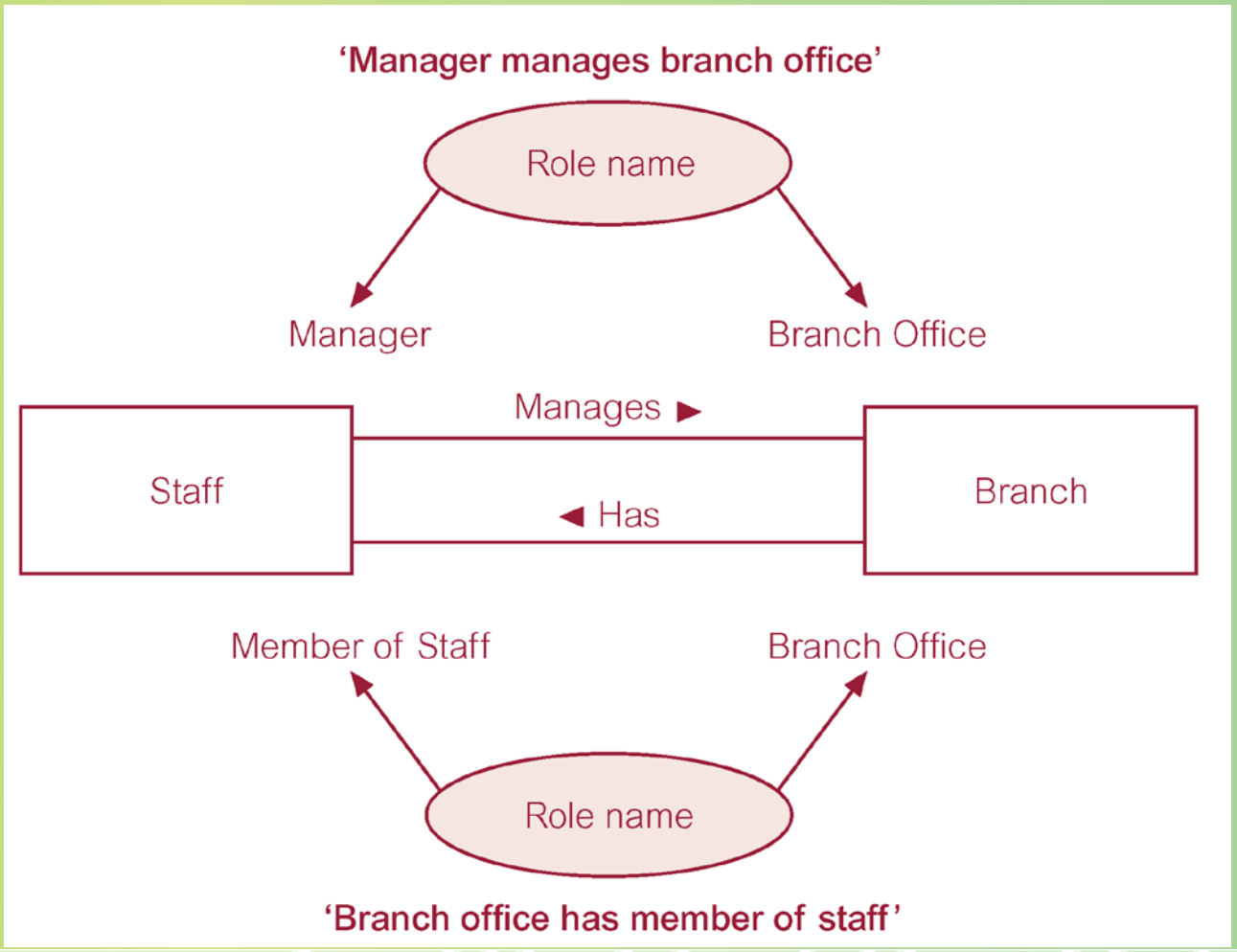
Relationship Types

- **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

- **Attribute**
  - **Property of an entity or a relationship type.**
- **Attribute Domain**
  - **Set of allowable values for one or more attributes.**

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.**

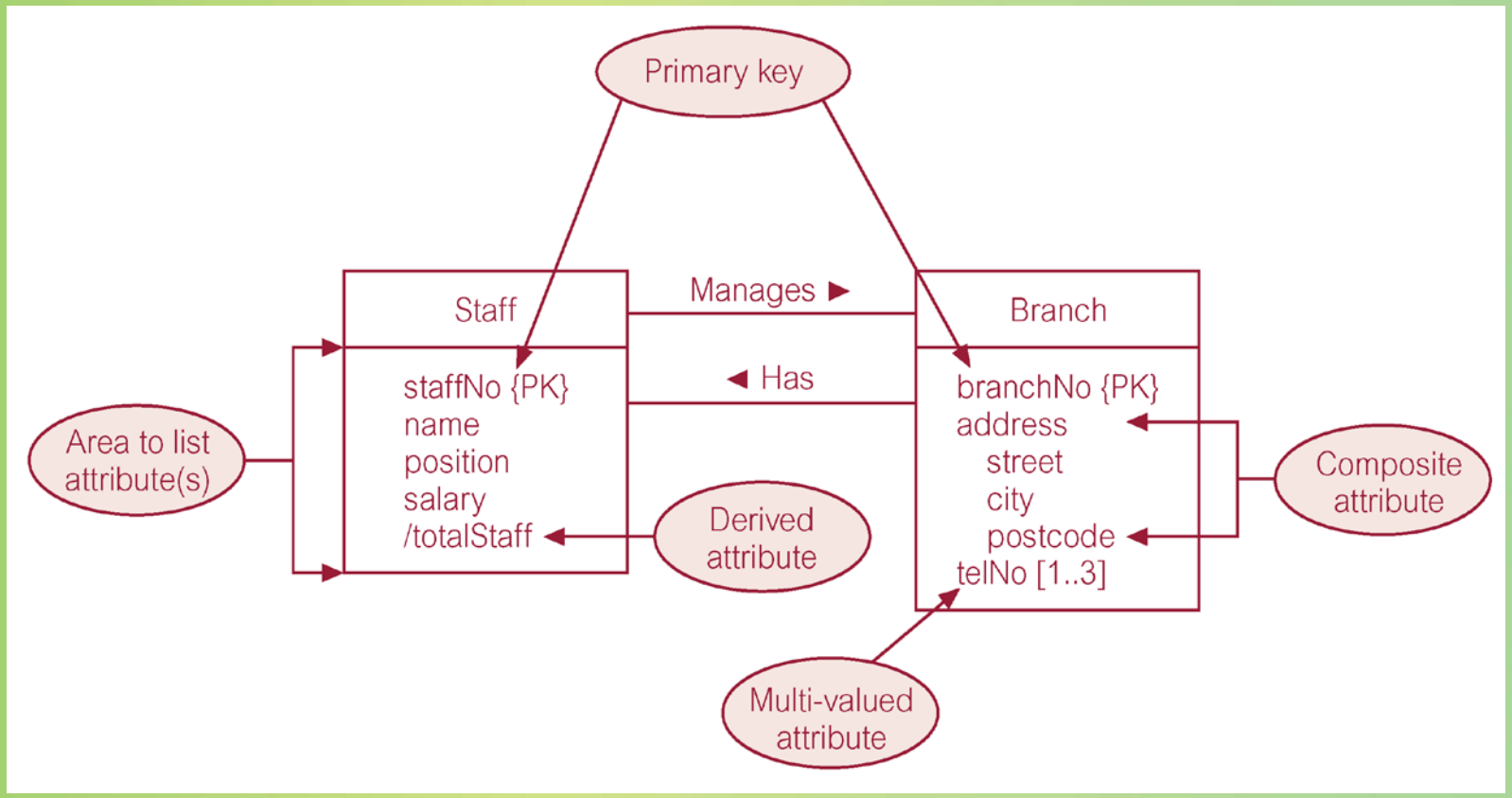
Attributes

- **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.**

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.**

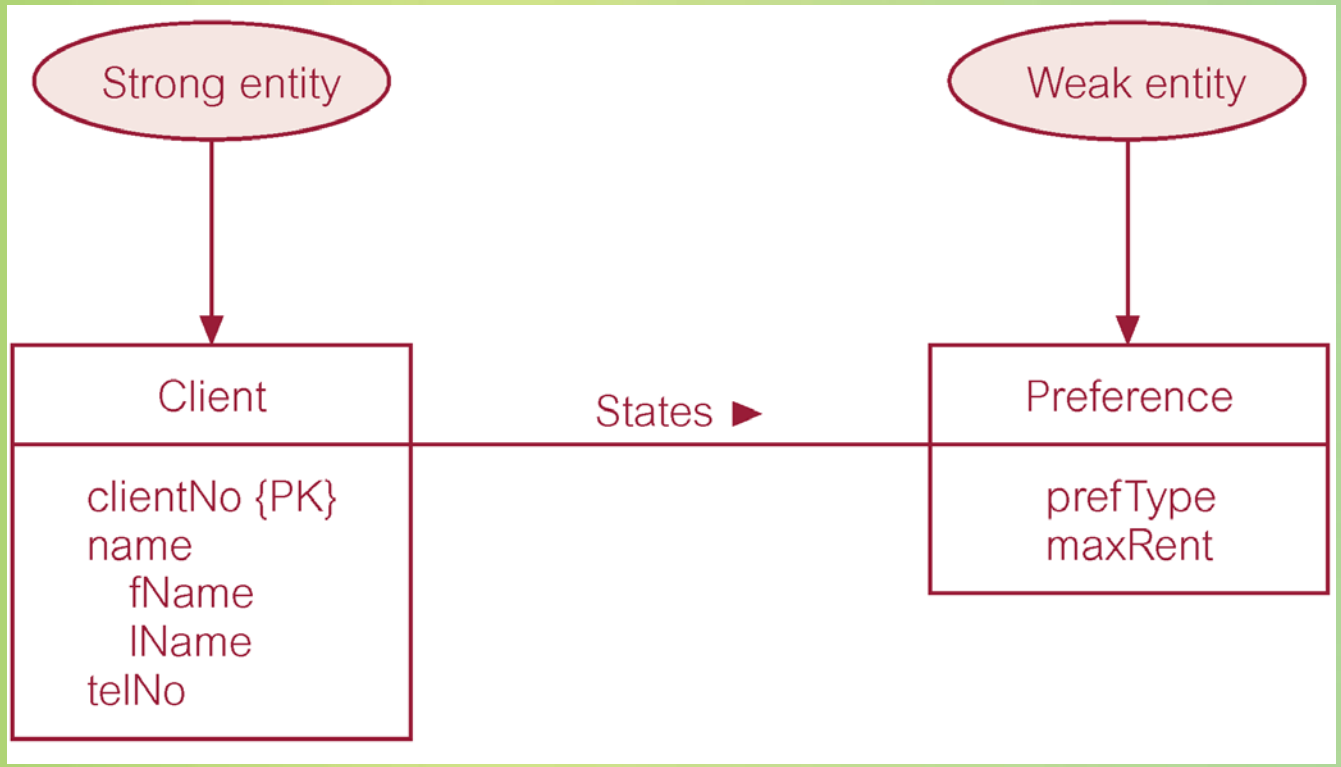
ER diagram of Staff and Branch entities and their attributes



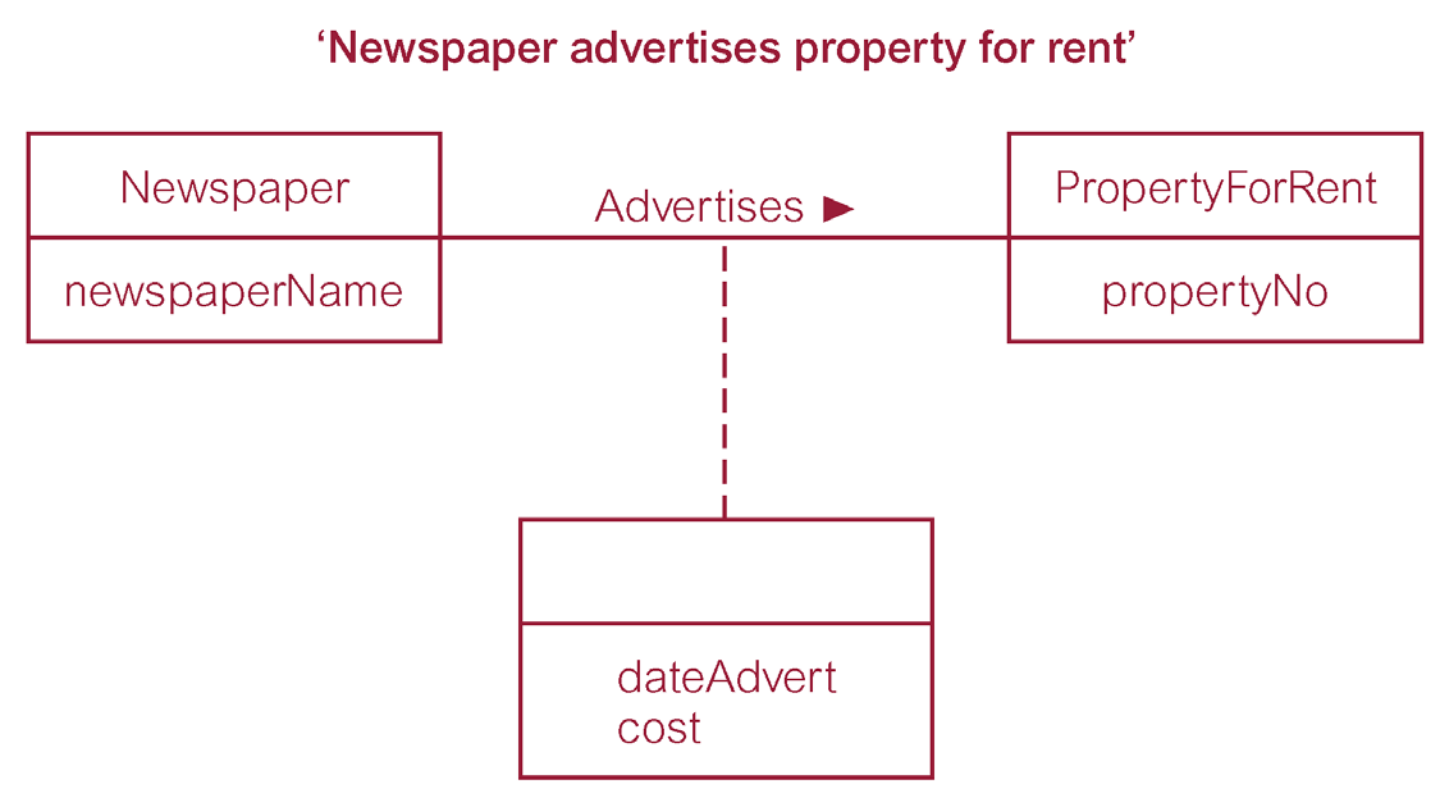
Entity Type

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

Strong entity type called Client and weak entity type called Preference



Relationship called *Advertises* with attributes



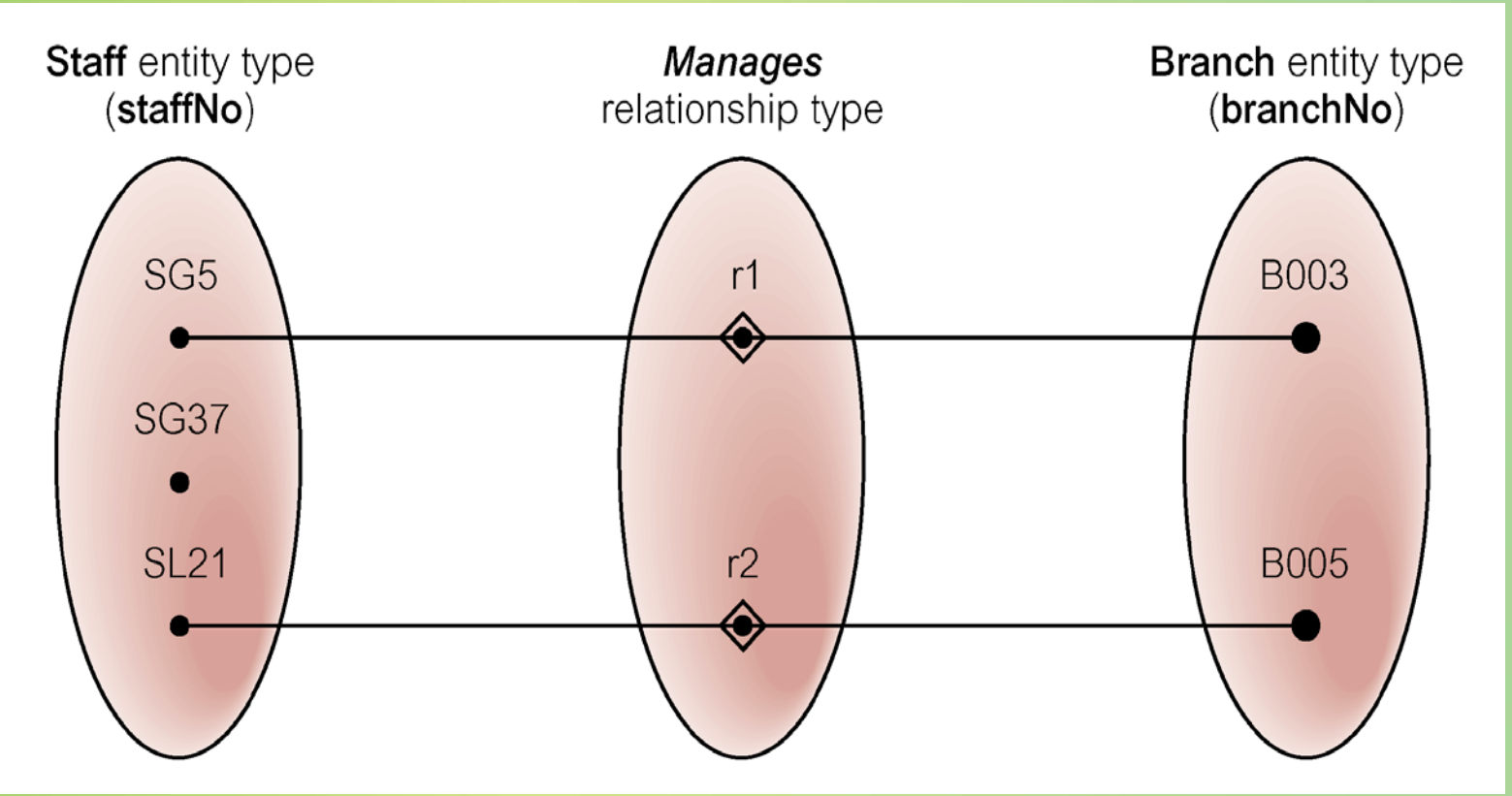
Structural Constraints

- **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.**

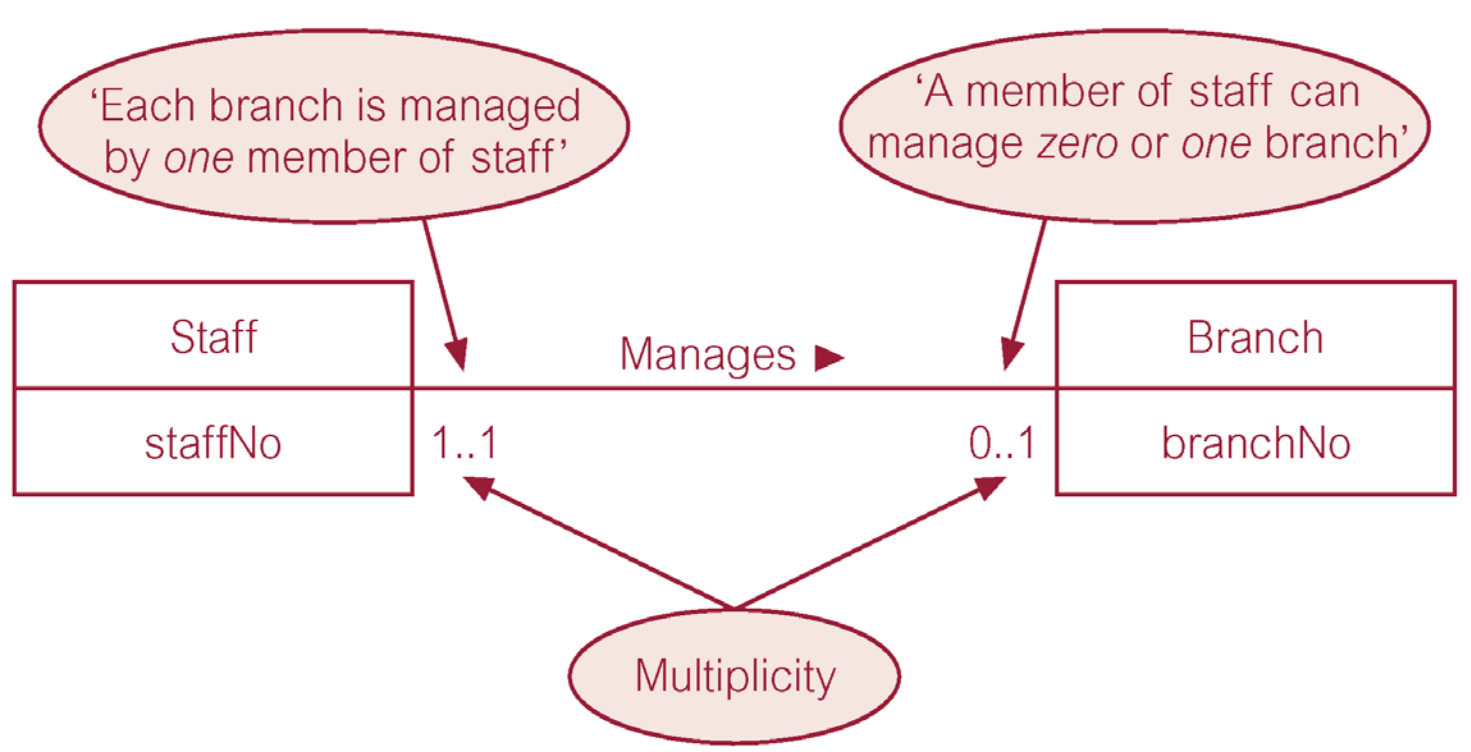
Structural Constraints

- 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 (*:*)

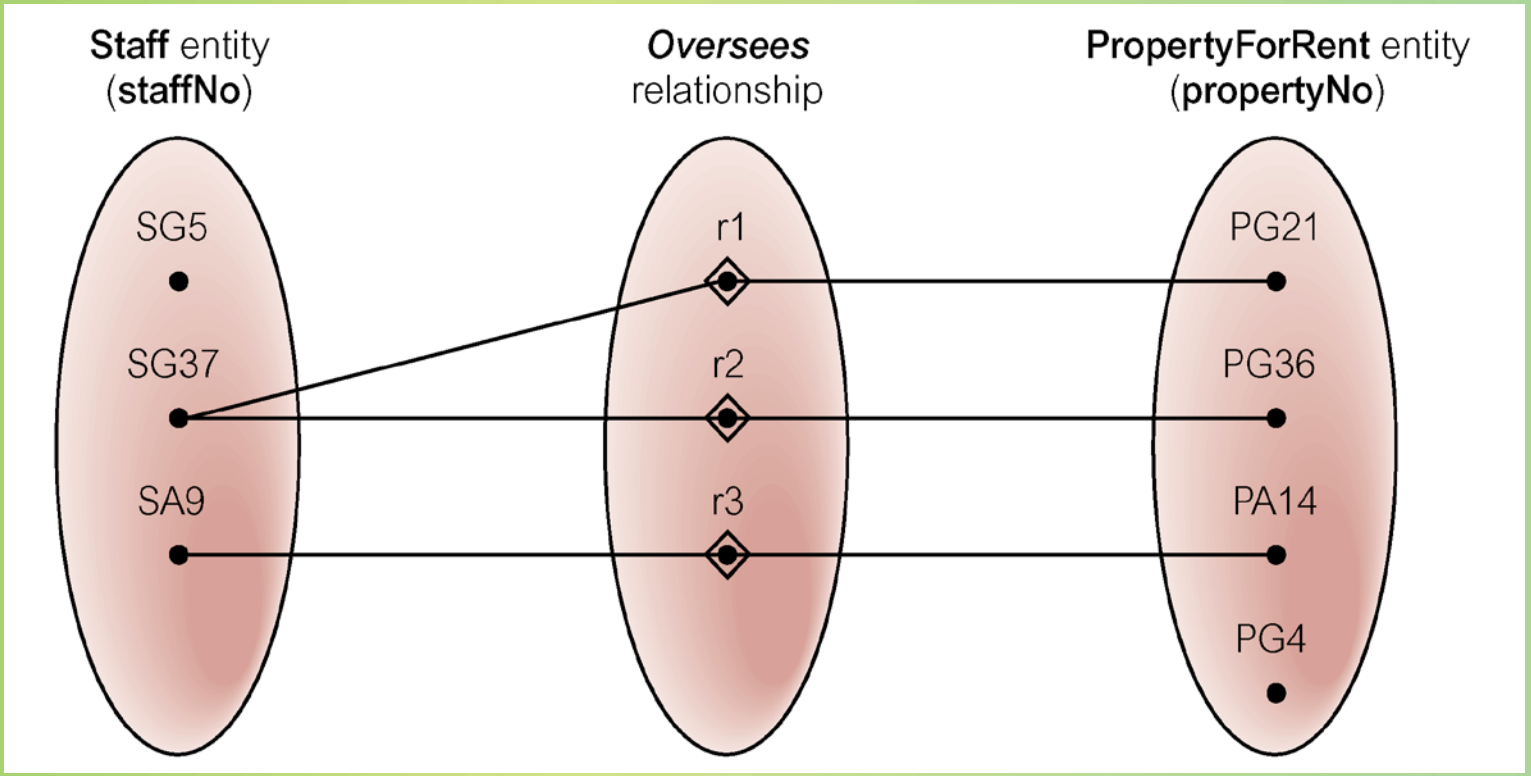
Semantic net of Staff *Manages* Branch relationship type



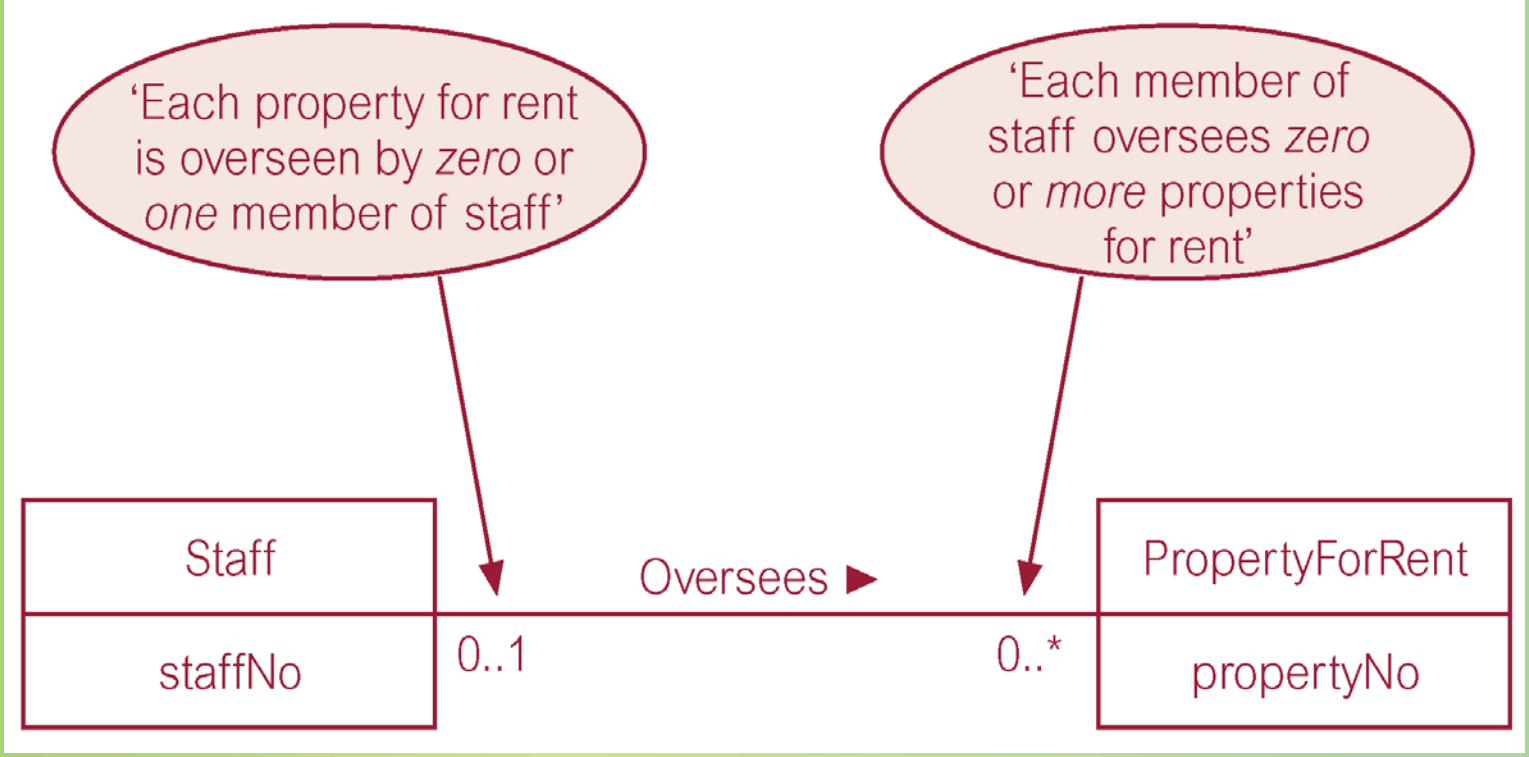
Multiplicity of Staff *Manages* Branch (1:1) relationship



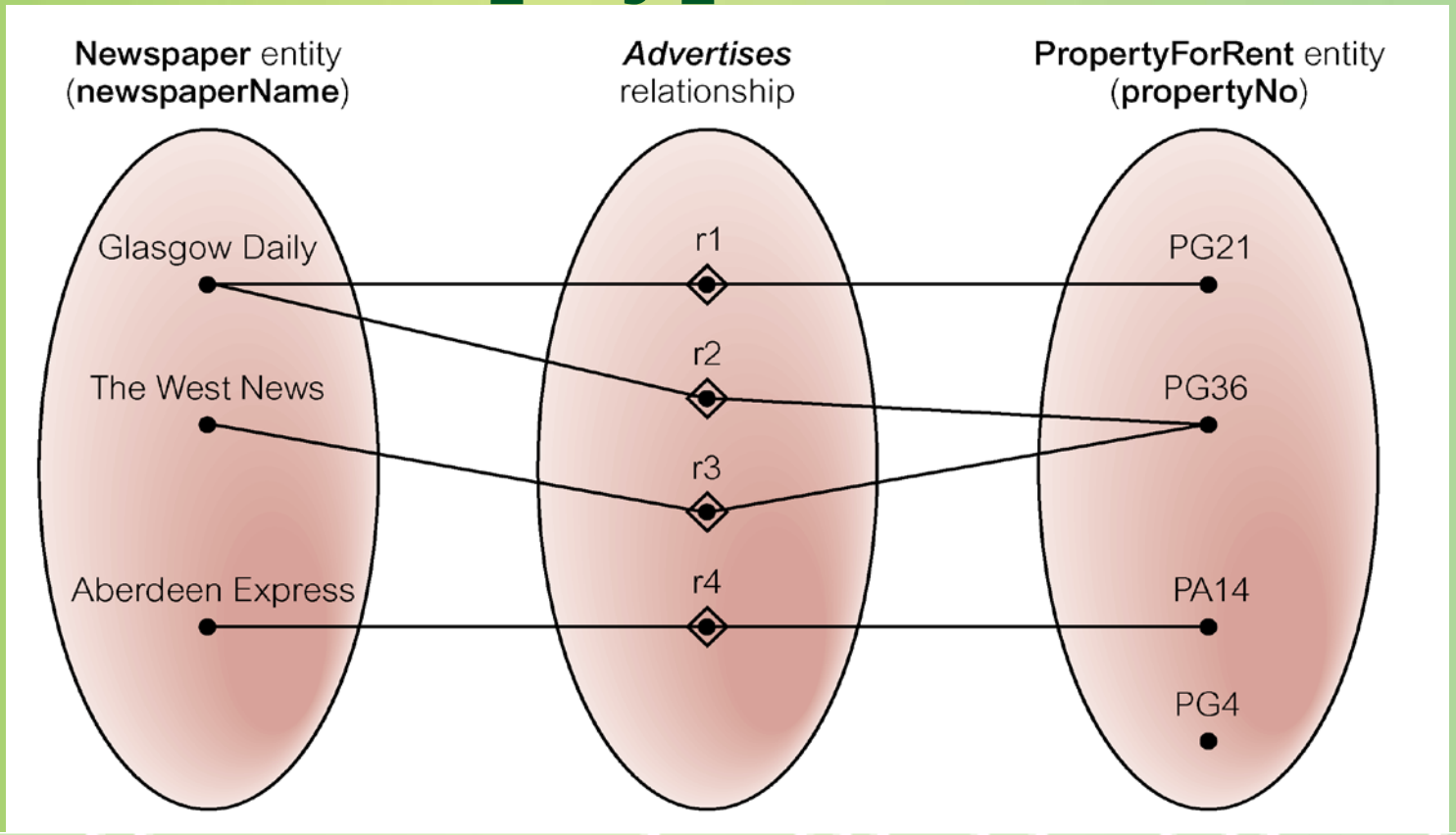
Semantic net of Staff *Oversees* PropertyForRent relationship type



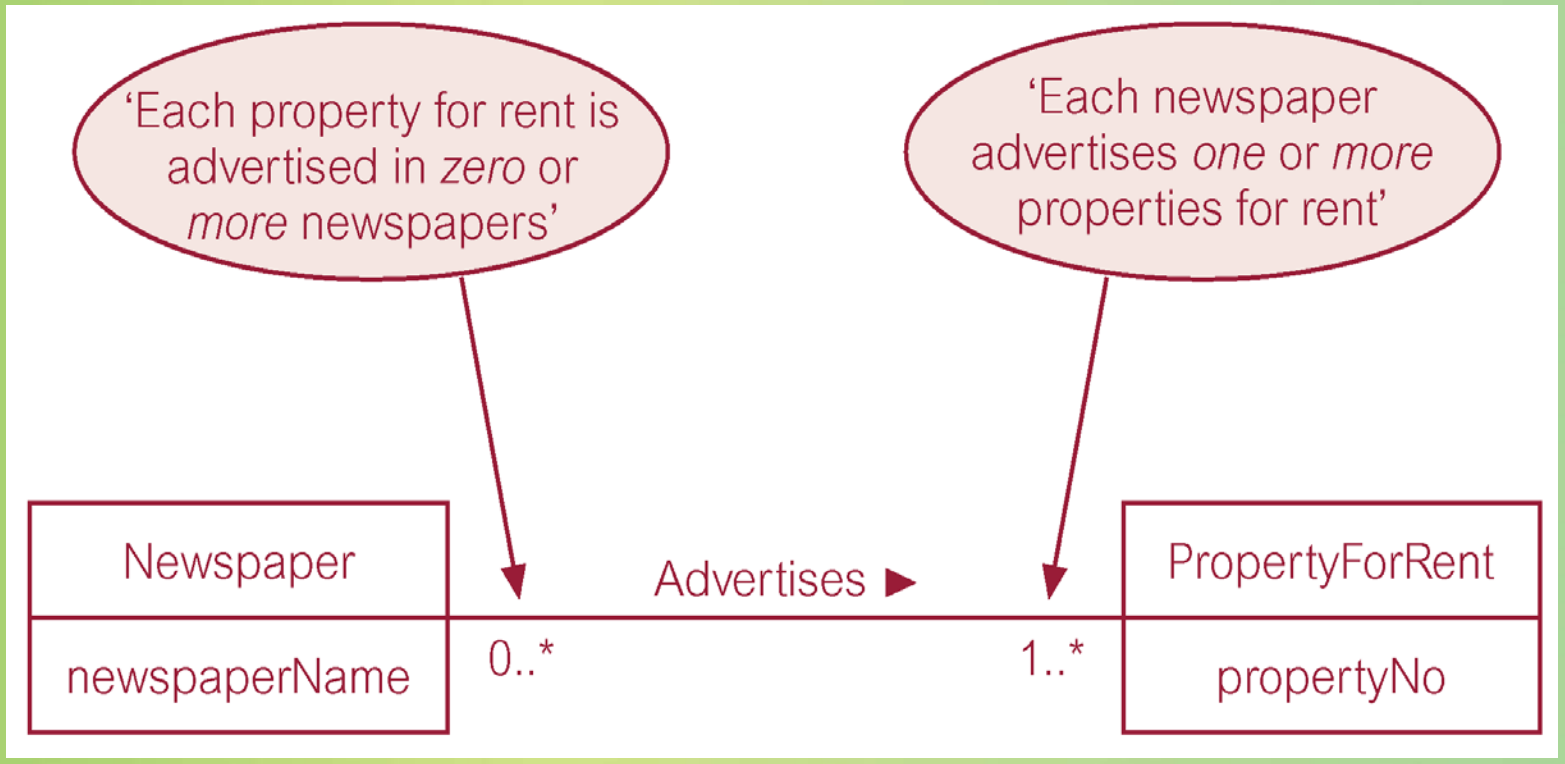
Multiplicity of Staff *Oversees* PropertyForRent (1:*) relationship type



Semantic net of Newspaper *Advertises* PropertyForRent relationship type



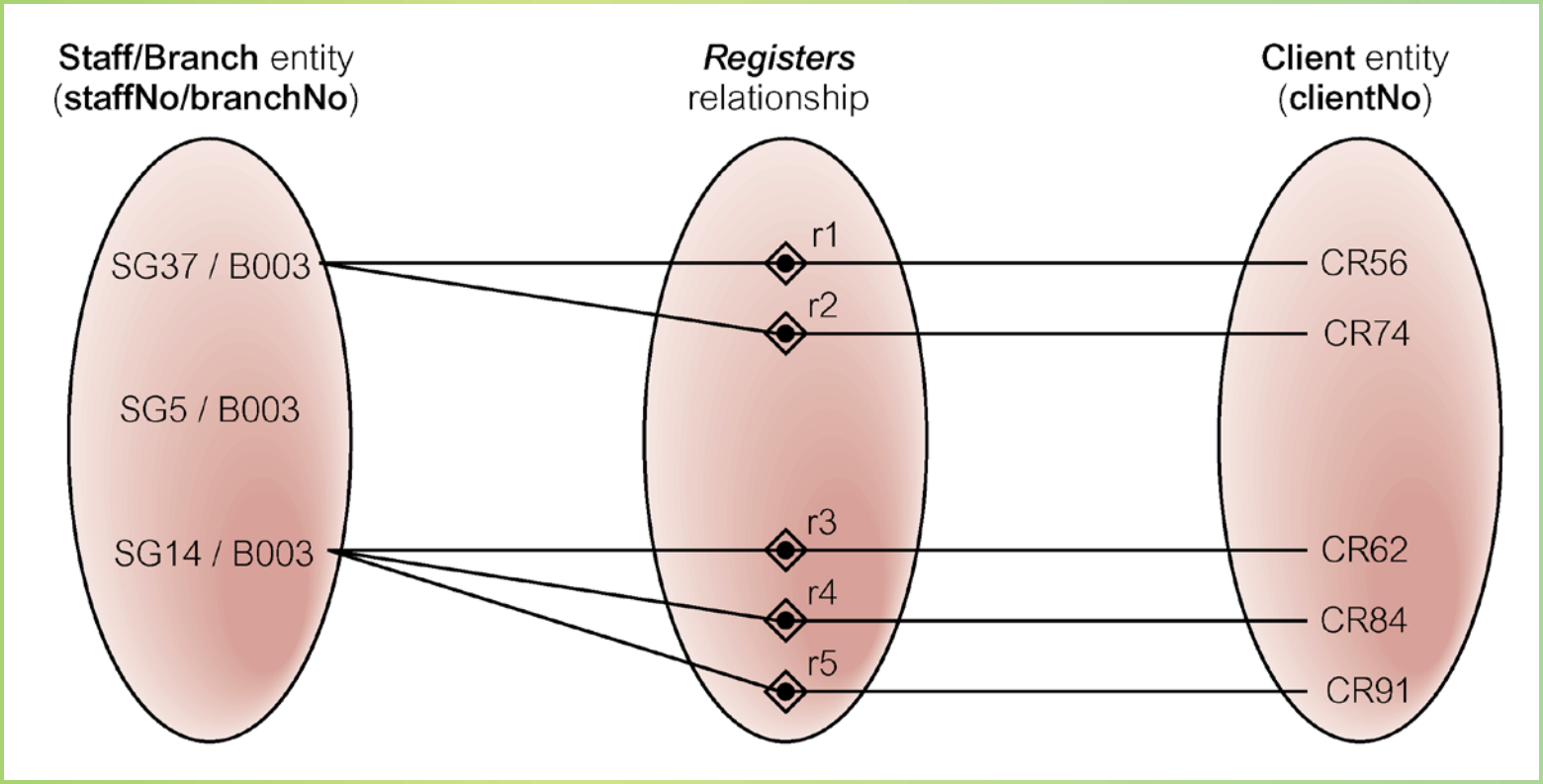
Multiplicity of Newspaper *Advertises* PropertyForRent (*:*) relationship



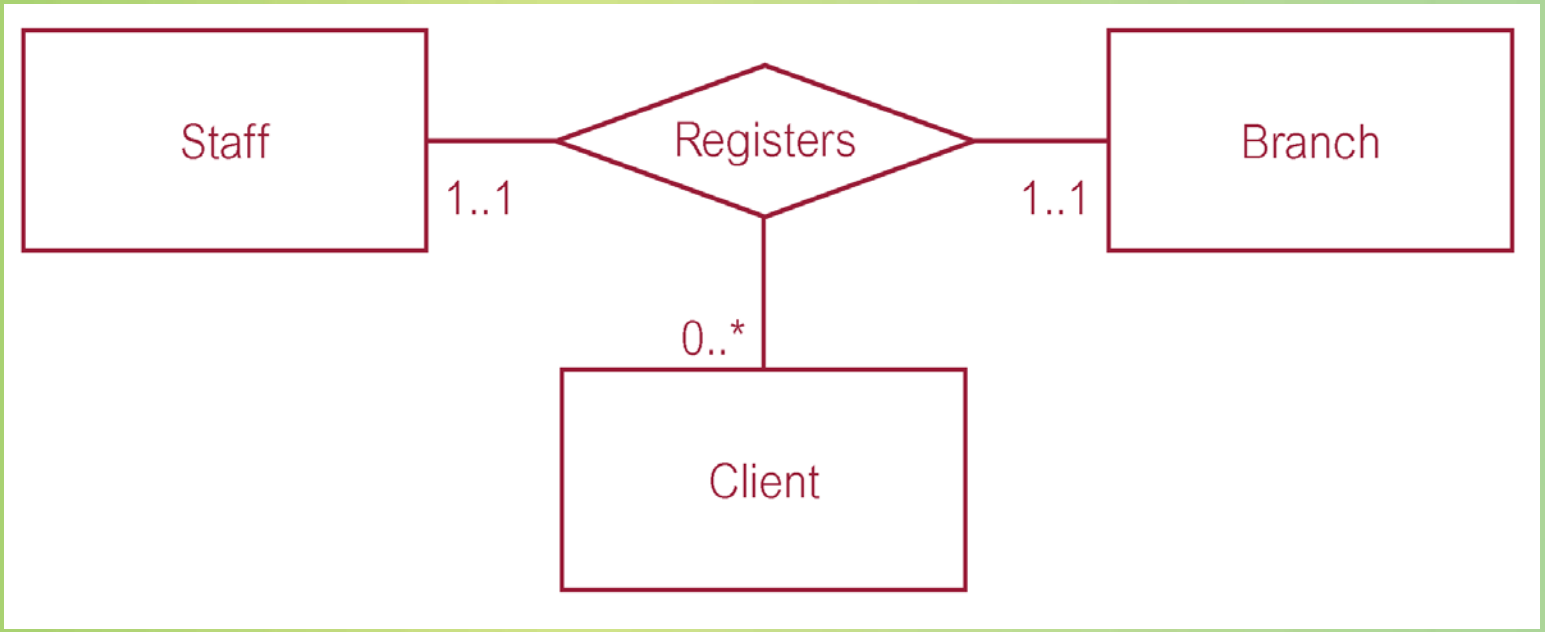
Structural Constraints

- **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.**

Semantic net of ternary *Registers* relationship with values for Staff and Branch entities fixed



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

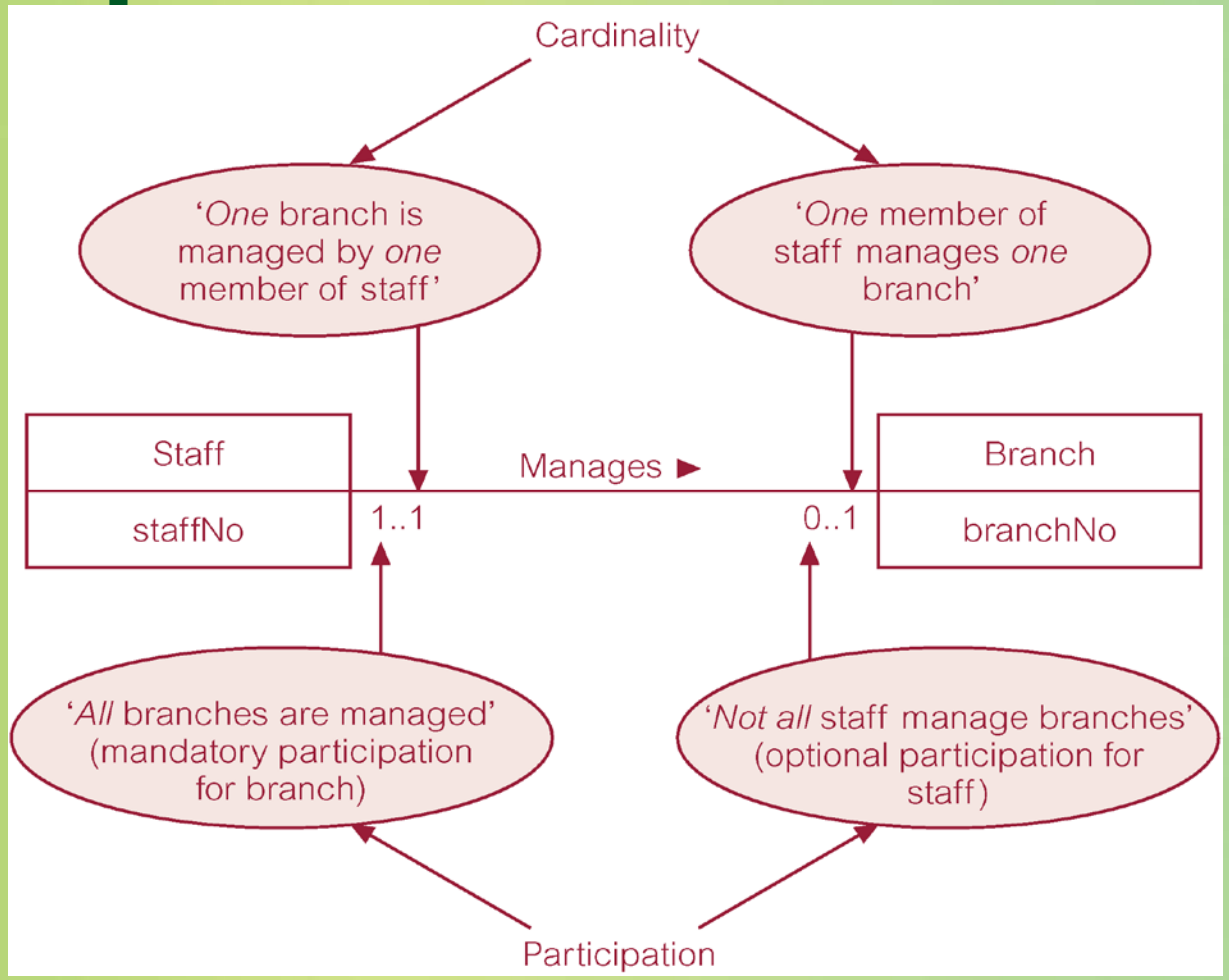
Structural Constraints

- **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 whether all or only some entity occurrences participate in a relationship.**

Multiplicity as cardinality and participation constraints



Problems with ER Models

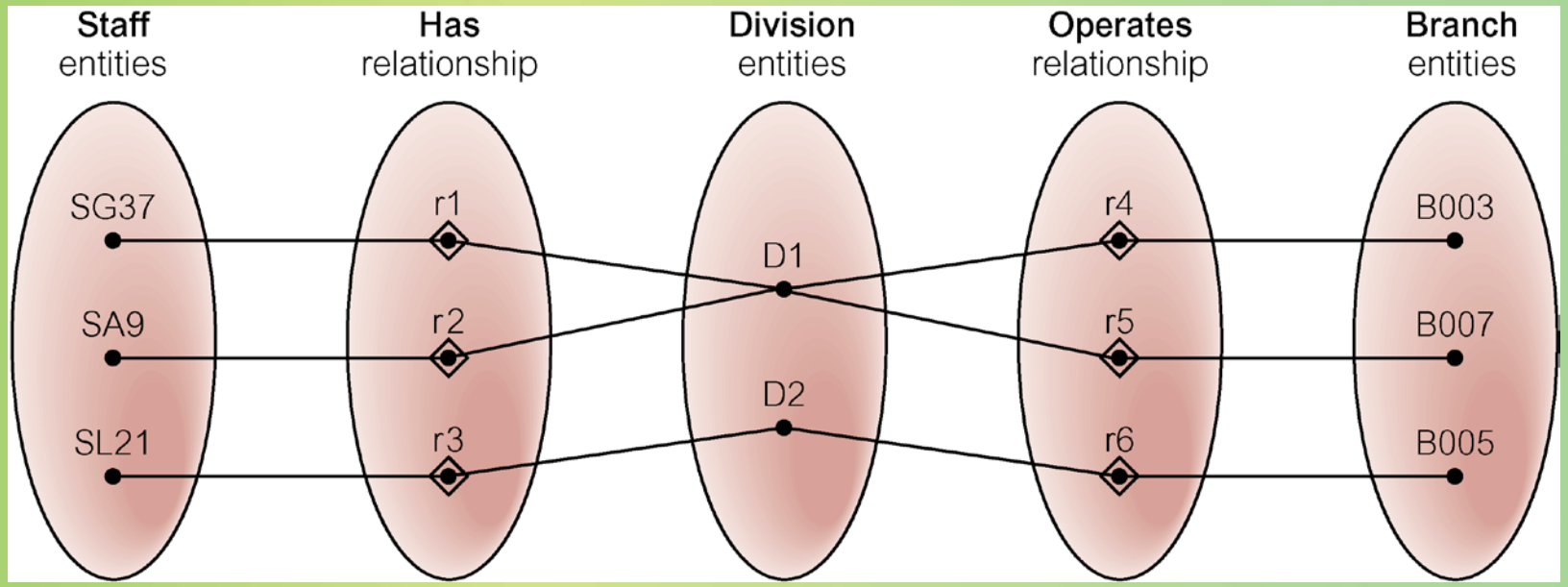
- **Problems may arise when designing a conceptual data model called *connection traps*.**
- **Often due to a misinterpretation of the meaning of certain relationships.**
- **Two main types of connection traps are called *fan traps* and *chasm traps*.**

Problems with ER Models

- **Fan Trap**
  - **Where a model represents a relationship between entity types, but pathway between certain entity occurrences is ambiguous.**
- **Chasm Trap**
  - **Where a model suggests the existence of a relationship between entity types, but pathway does not exist between certain entity occurrences.**

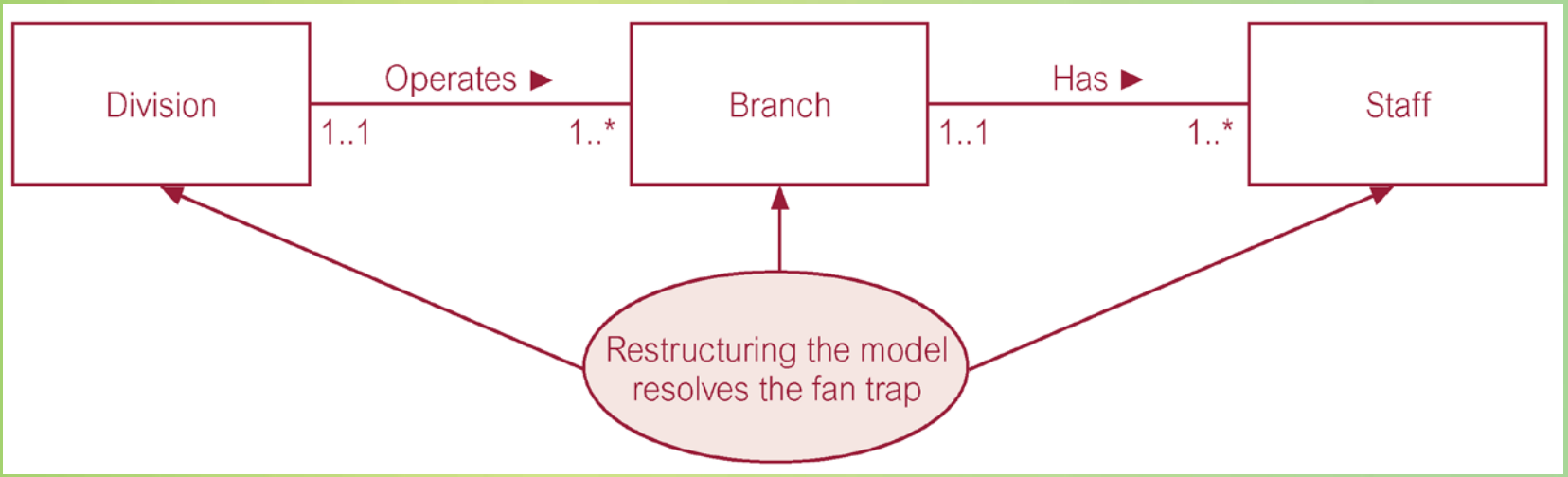
An Example of a Fan Trap

Semantic Net of ER Model with Fan Trap

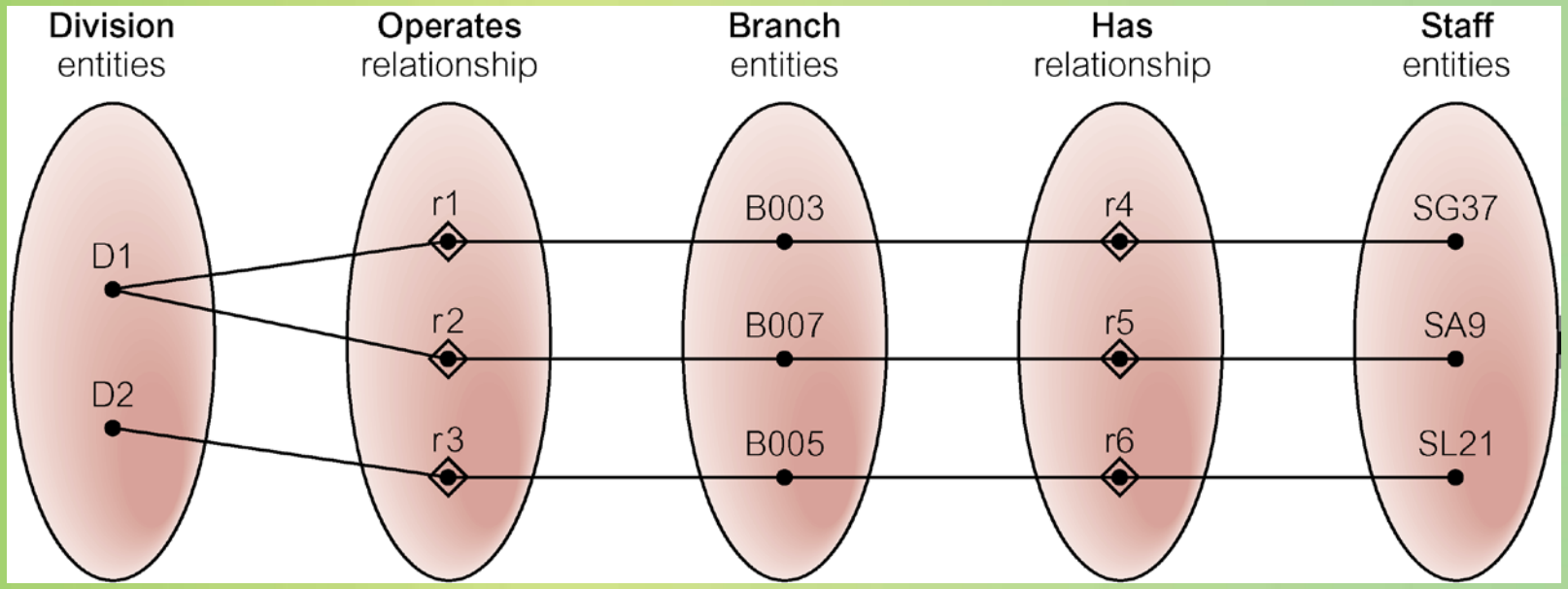


- **At which branch office does staff number SG37 work?**

Restructuring ER model to remove Fan Trap



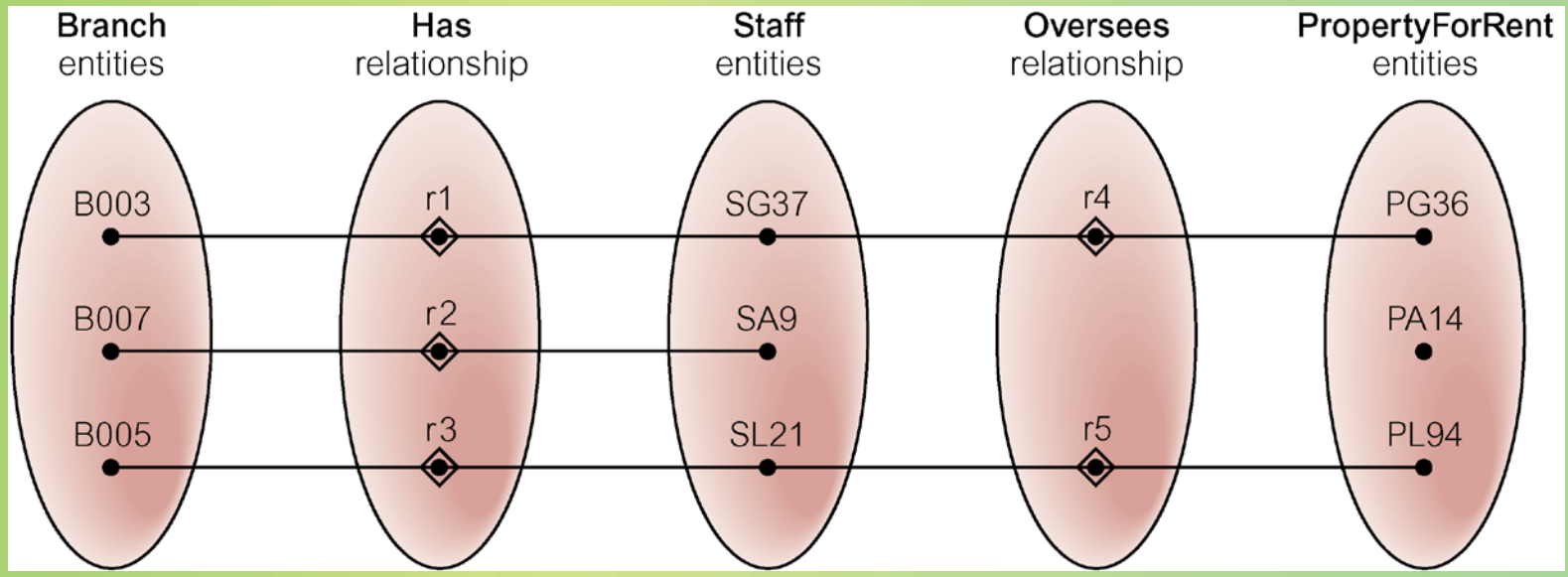
Semantic Net of Restructured ER Model with Fan Trap Removed



- **SG37 works at branch B003.**

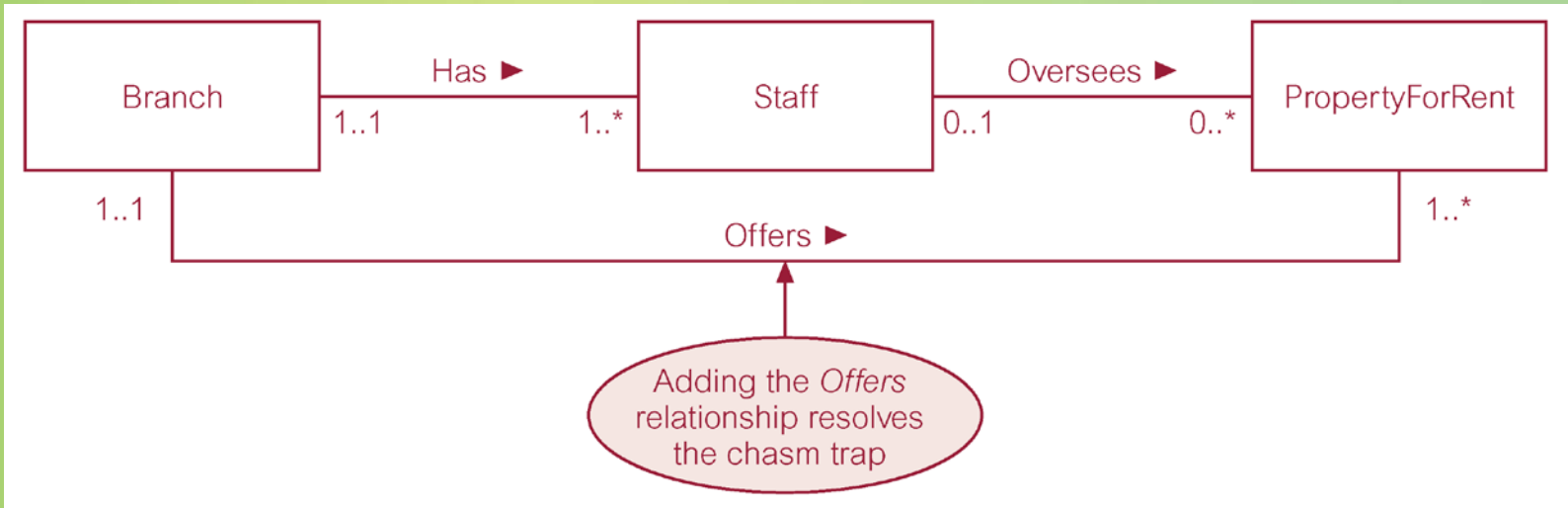
An Example of a Chasm Trap

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