# LECTURE 3

# ENTITY-RELATIONSHIP MODELING

Muhammad Hamiz Mohd Radzi
Faiqah Hafidzah Halim

# Content

- Entity Types
- Relationship Types
- Attributes
- Strong and Weak Entities
- Attributes on Relationship
- Structural Constraints
- Enhanced ER Modeling
- Specialization/Generalization

# Objectives

- At the end of this lesson, you should be able to:
  - *Define entity, entity types and entity occurrence*
  - *Describe relationship types, occurrence, degree of relationship types, and recursive relationship.*
  - *Explain attribute, attribute domain, simple, composite, single, multivalued and derived attributes.*
  - *Describe the strong and weak entities.*
  - *Explain attribute on relationship*

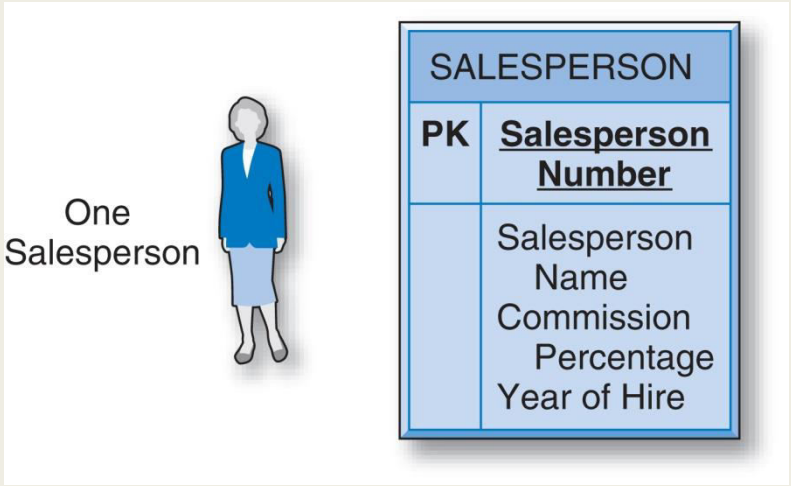
- *Explain the multiplicity, cardinality and participation*
- *Describe fan and chasm trap*
- *Define the enhance ER Modeling*
- *Explain the concept of superclass/subclass, its relationship, specialization and generalization process, and the constraint of specialization/generalization.*

# The entity relationship model

- ERM is existed as to explore the different ways that entities can relate to each other as they always do in the real world.
- ERM also is used to devise a way of recording, of diagramming, the entities and the ways in which they interrelate in the business environment.
- The things that you need to know in order to construct the ERM are:
  - *Entity*
  - *Attribute*

# Entity

- Entity - a “**thing**” or “**object**” in our environment that we want to keep track of.
- Entity set - A **collection of entities** of the same type (e.g., all of the company’s employees).
- Entity type - A group of objects with the same properties, which are identified by the enterprise as having an independent existence.
- Entity occurrence - A uniquely identifiable object of an entity type.




- Entity is in rectangular shape
- Name of entity is in caps above the separator line.
- For example:
  - *Entity type = SALESPERSON*

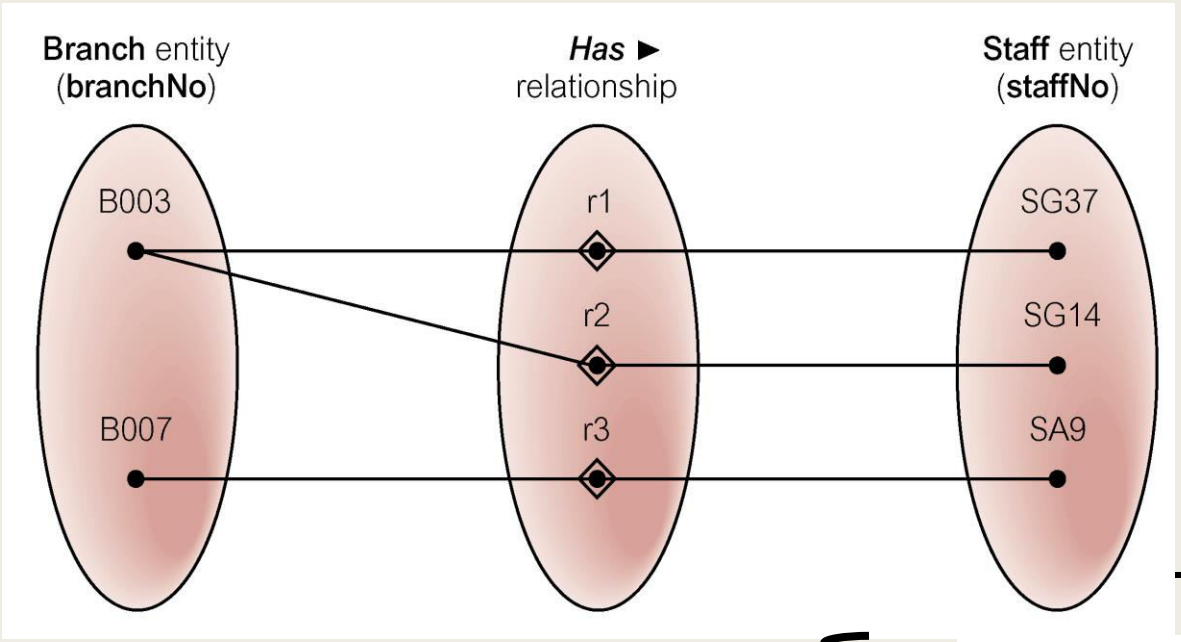
# Relationship Types

**Relationship Type**

- Set of meaningful associations among entity types. Each relationship type is given a name that describes its function.

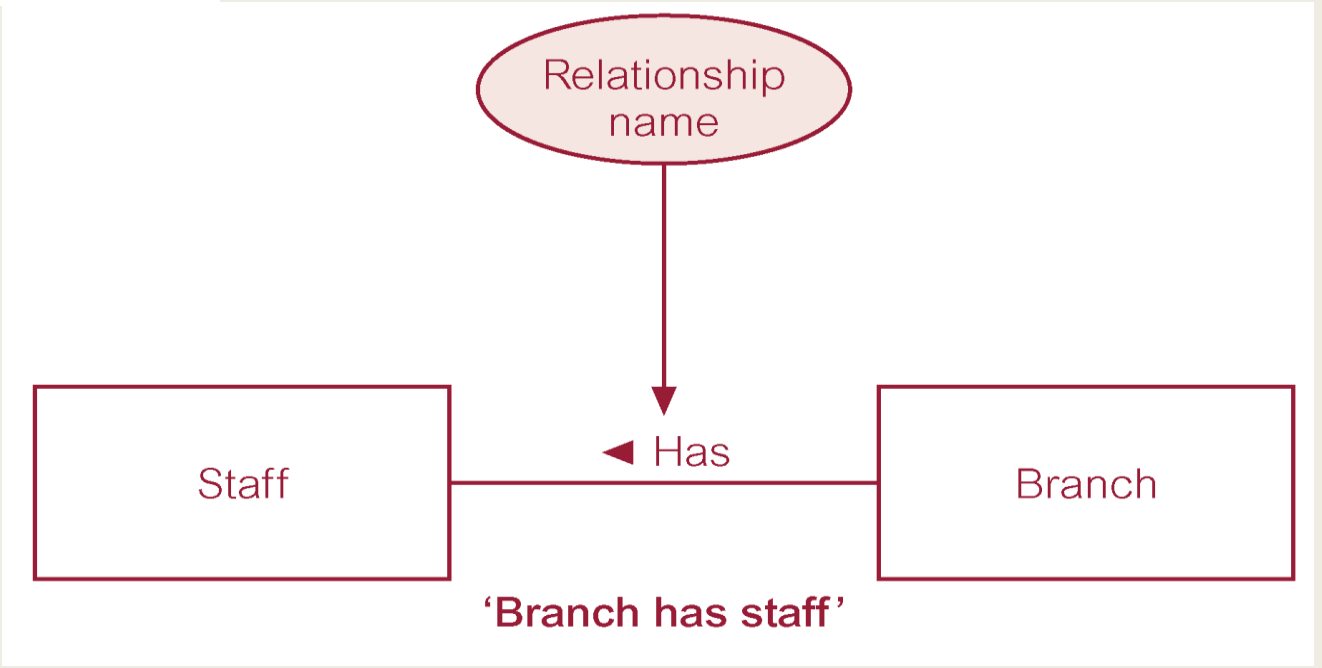
**Relationship occurrence**

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



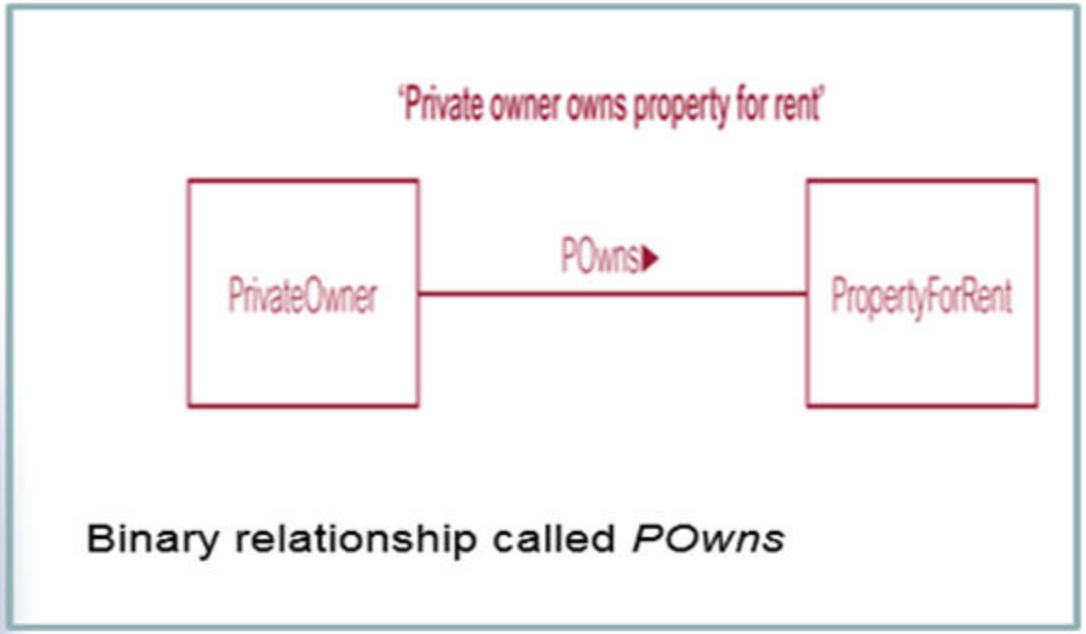

**Semantic Net View**

**Diagrammatic View**

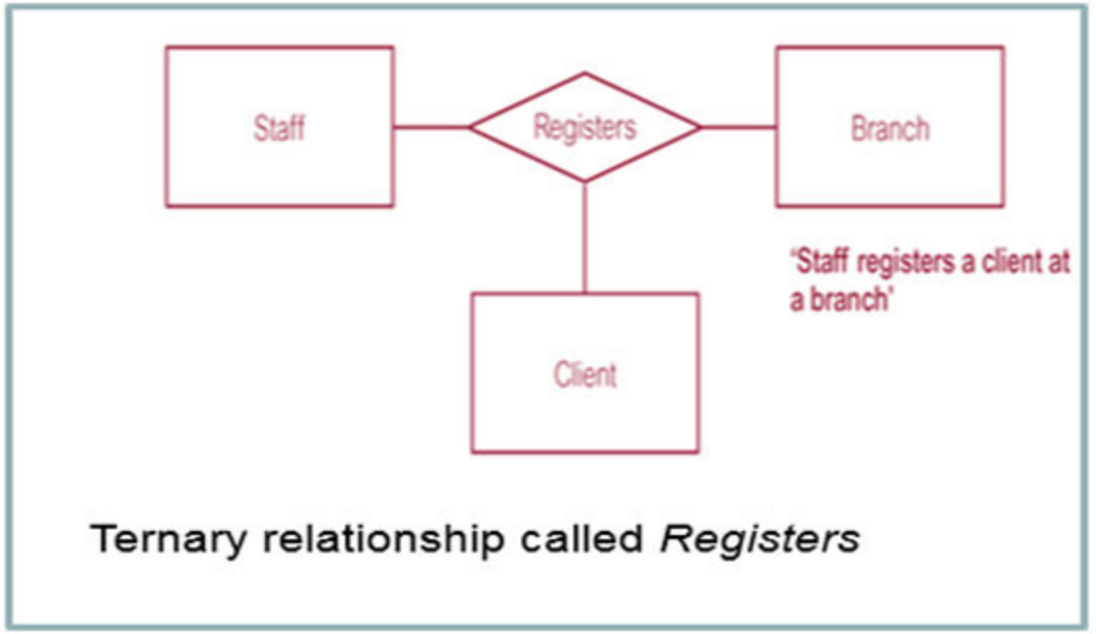


# Degree of Relationship

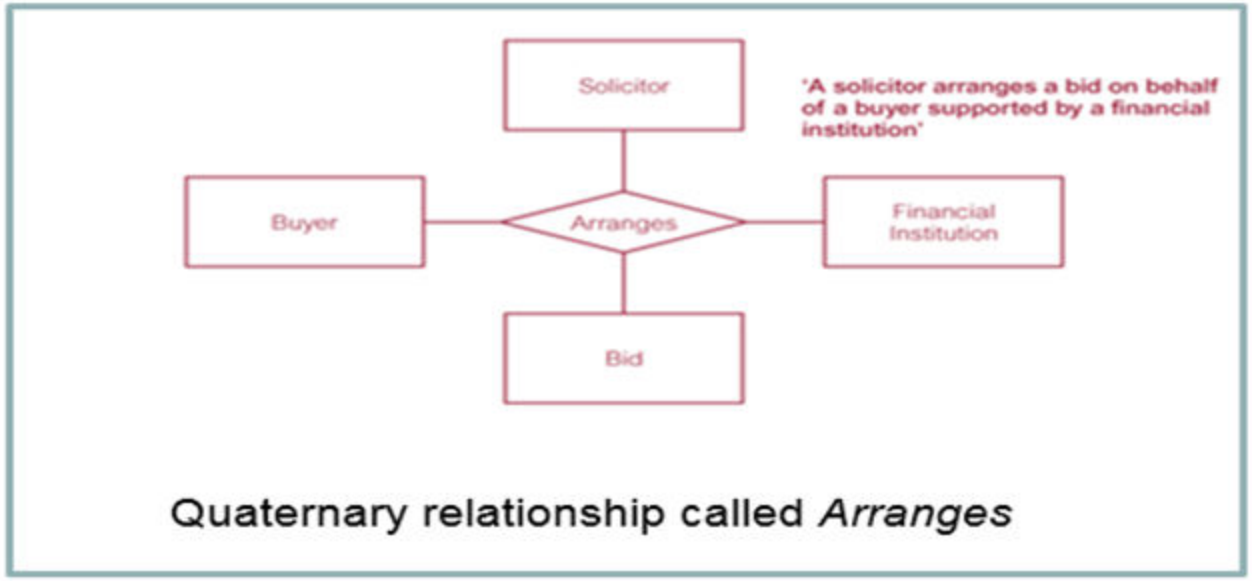
- **It is defined as** number of participating entities in  relationship.
- **Relationship of degree :**
  - *two is binary*
  - *three is ternary*
  - *four is quaternary.*




Binary relationship called *POwns*



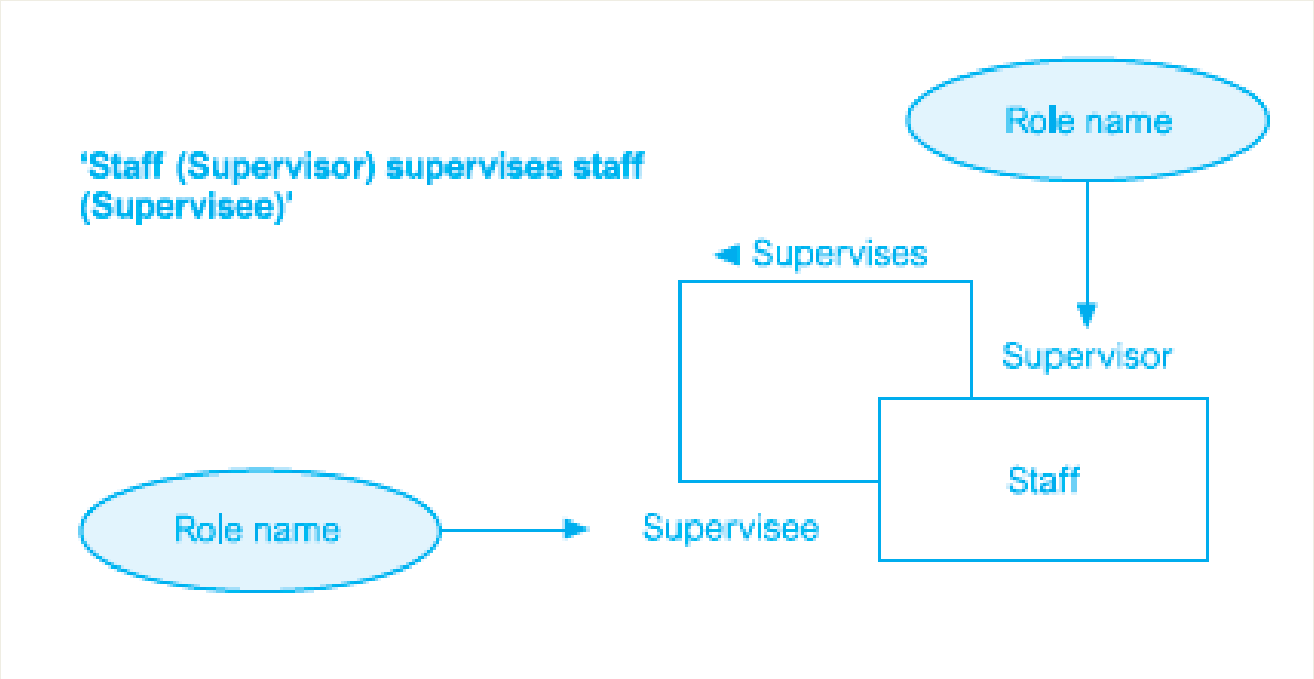

Ternary relationship called *Registers*



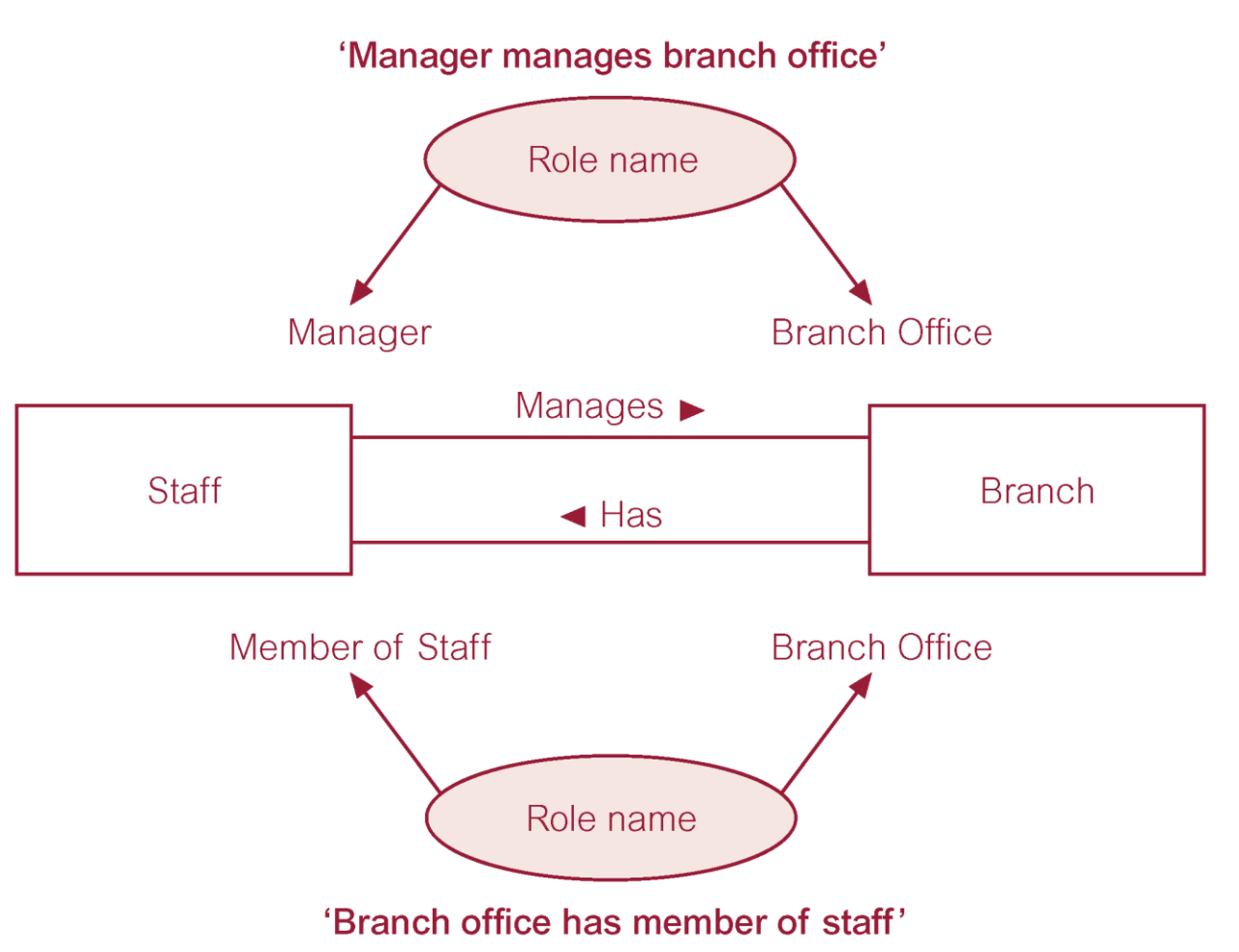

Quaternary relationship called *Arranges*

# Recursive Relationship

- Recursive relationship - A relationship type where the same entity type participates more than once in different roles.



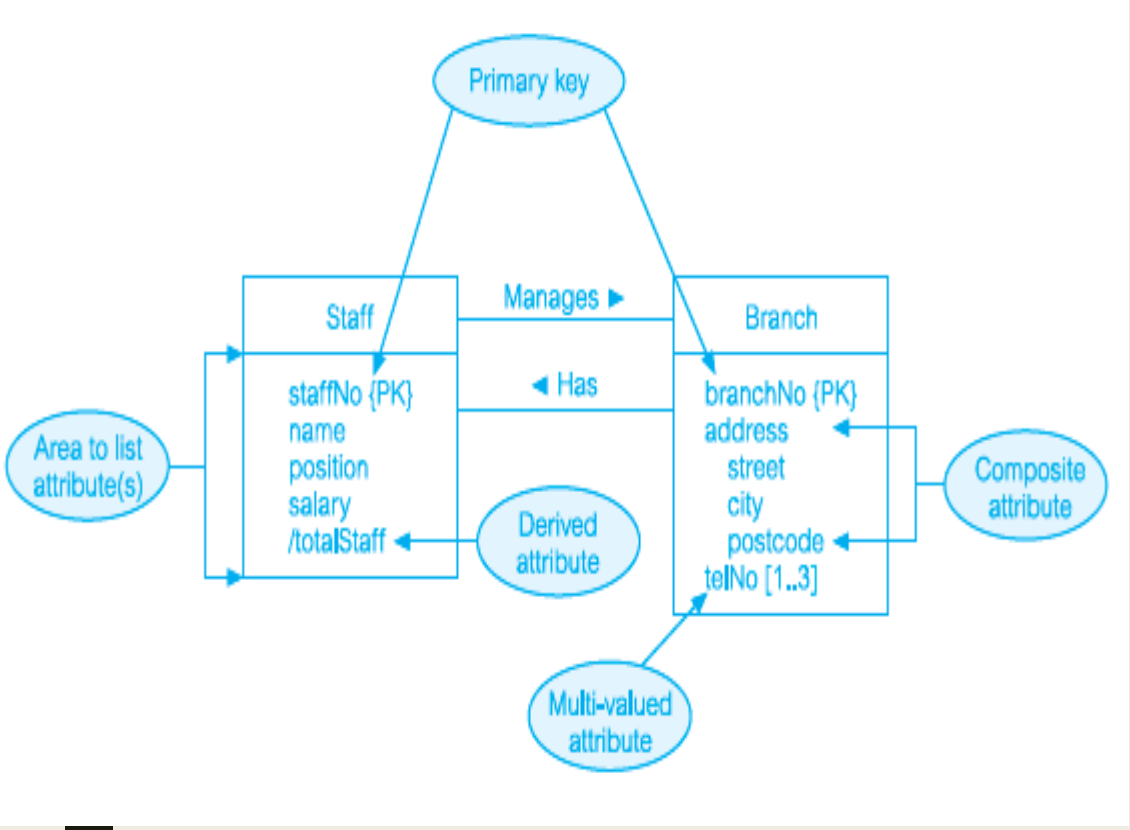
# Entities associated through two distinct relationships with role names



# Attribute

- Attribute - a property of, a **characteristic** of, or a fact that we know about an entity.
- Some attributes have unique values within an entity set which we called as Primary Key (PK)
- Attribute domain - The set of **allowable values** for one or more attributes.
  - *Each attributes have its own domain of values, for example, Room Number must be an Integer range from 1000-1999*

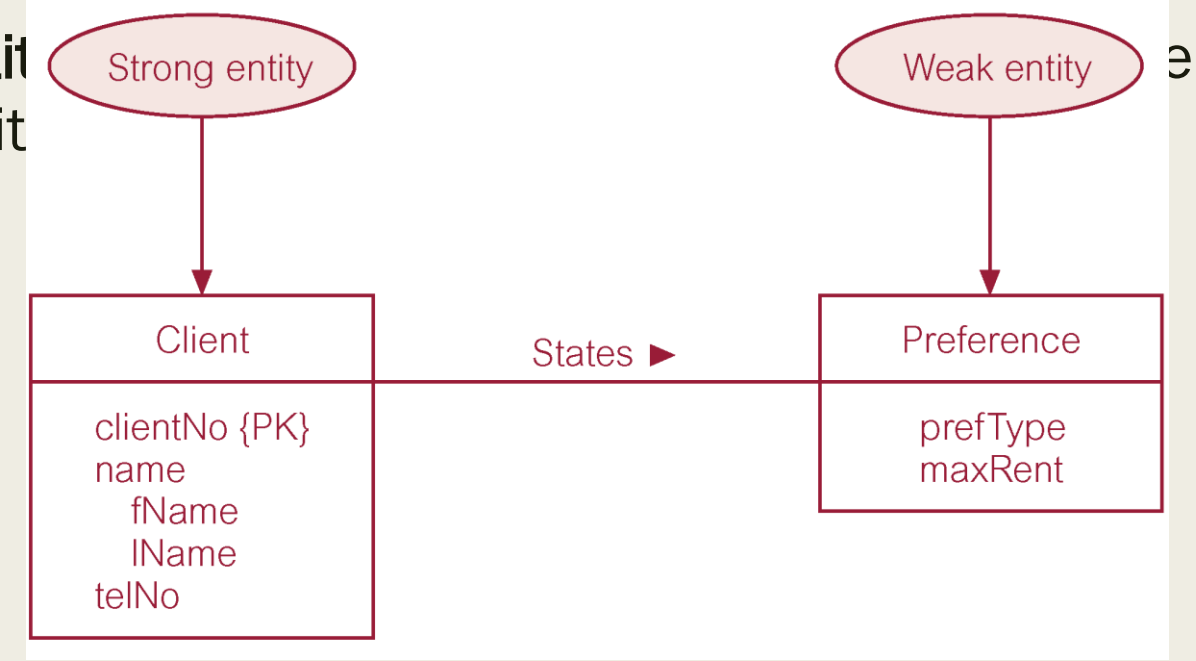
- Simple attribute - An attribute composed of a **single component** with an independent existence.
  - *It cannot be further subdivided into smaller components. For example Salary.*
- Composite attribute - An attribute composed of **multiple components**, each with an independent existence.
  - *Some attributes can be further divided to yield smaller components. For example, address can be divided into Street, City, Postcode.*
- Single-valued attribute - An attribute that holds a **single value** for each occurrence of an entity type.
  - *For example, Salesperson Number. Only one value can be attached to the attribute.*



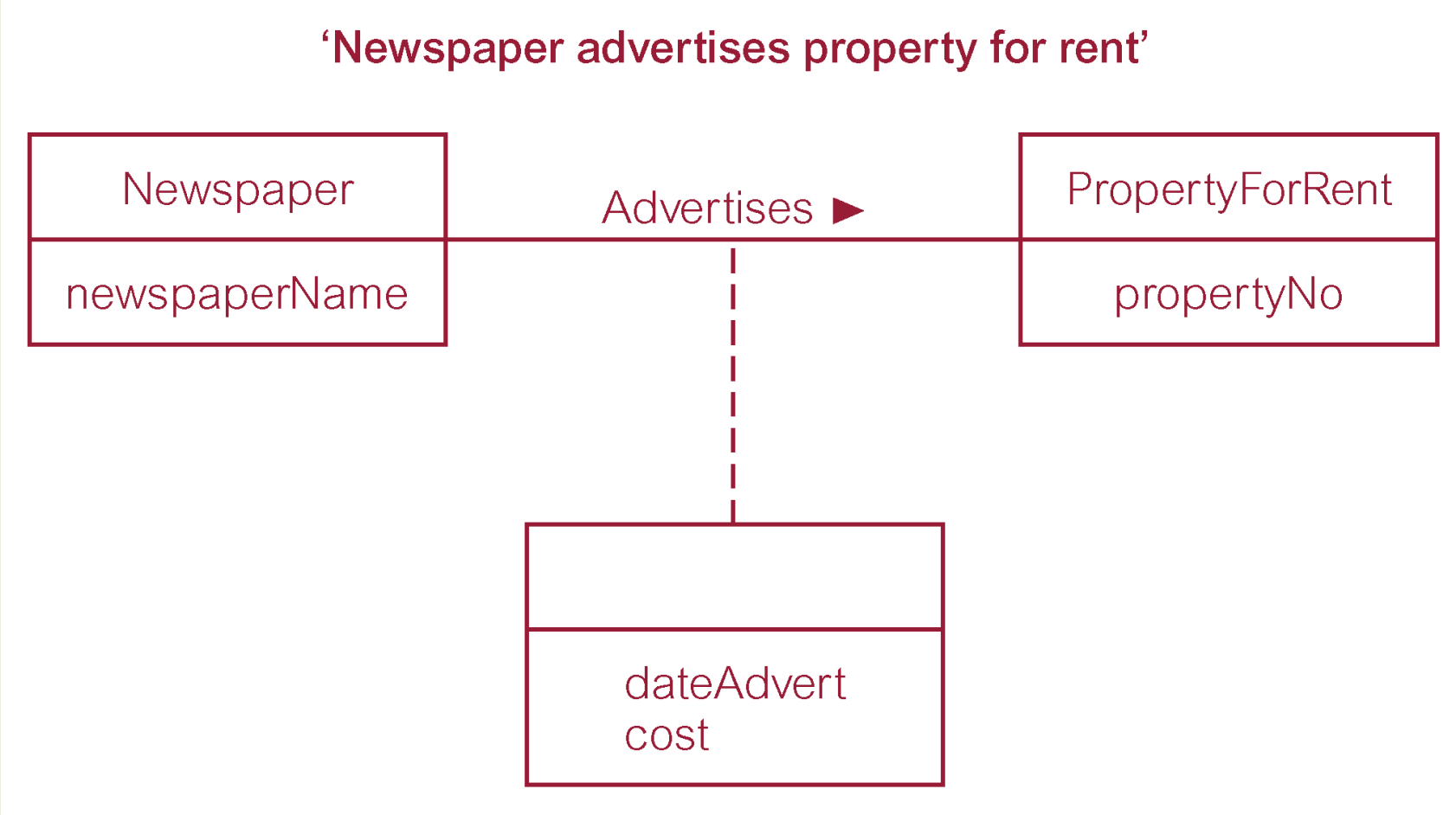

- Multi-valued attribute - An attribute that holds **multiple values** for each occurrence of an entity type.
  - *for example, branch number B003 has telephone numbers 0141-339-2178 and 0141-339-4439*
- Derived attribute - An attribute that represents a value that is **derivable from the value of a related attribute** or set of attributes, not necessarily in the same entity type.
  - *For example, the total number of staff (totalStaff) attribute of the Staff entity type can be calculated by counting the total number of Staff entity occurrences.*

# Strong and Weak Entity Types

- **Strong entity type:** An entity type that is *not* existence-dependent on some other entity type.
- **Weak entit** ...endent on some other entit...



# Attributes on Relationship

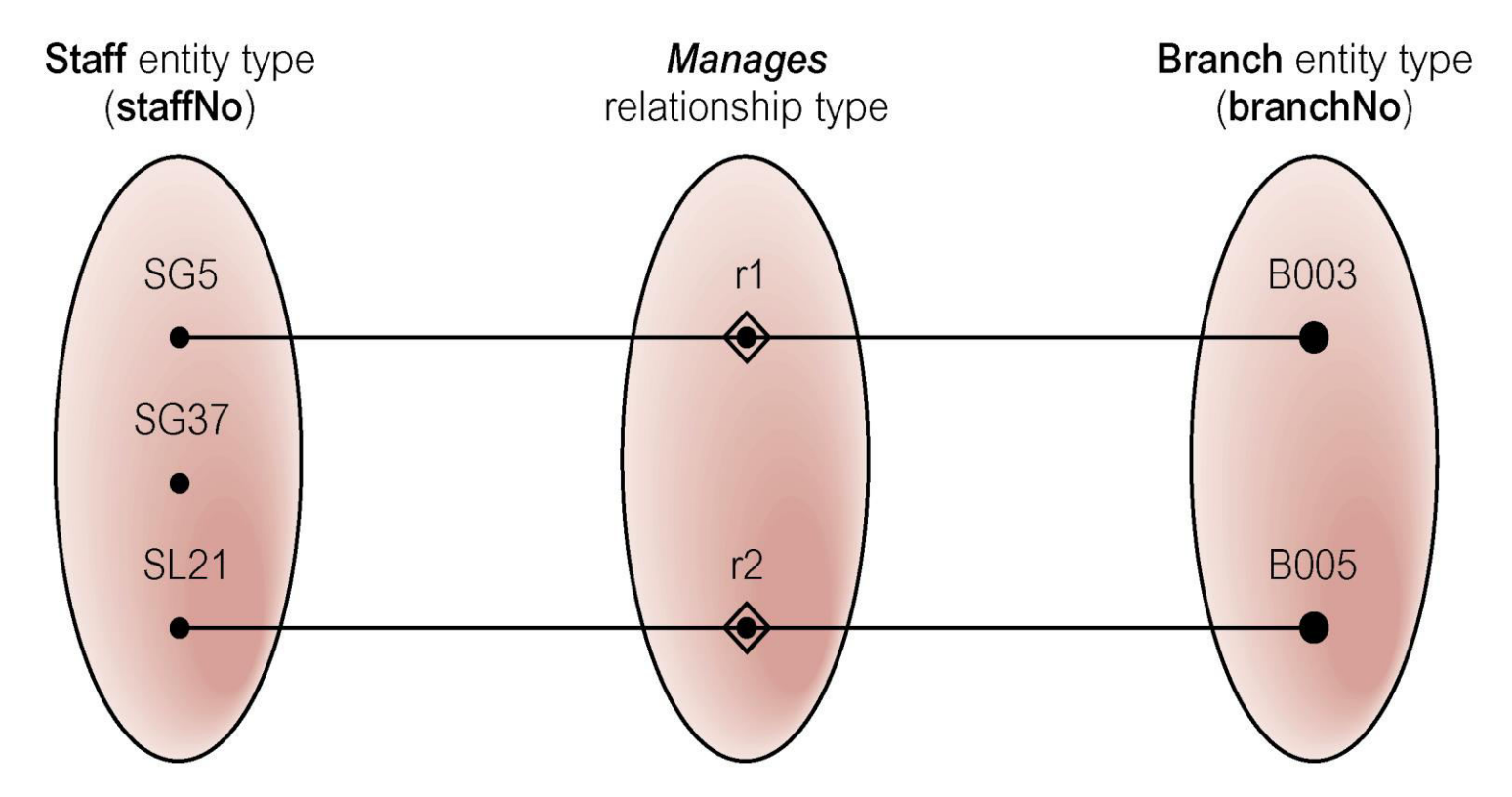


# Structural Constraints

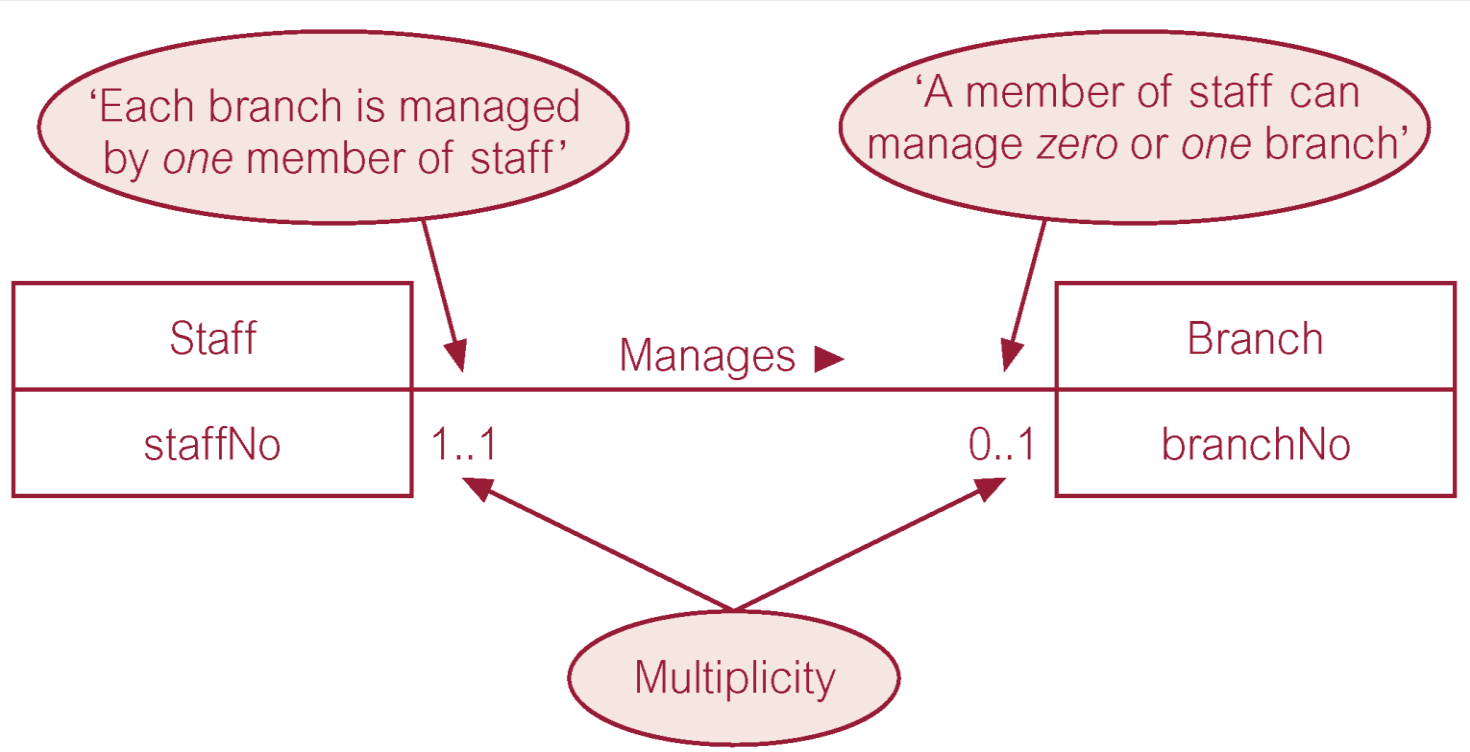
- Main type of constraint on relationships is called ***multiplicity***.
- **Multiplicity** - number (or range) of possible occurrences of an entity type that may relate to a single occurrence of an associated entity type through a particular relationship.
- Represents policies (called *business rules*) established by user or company.
- 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 (*:*)*

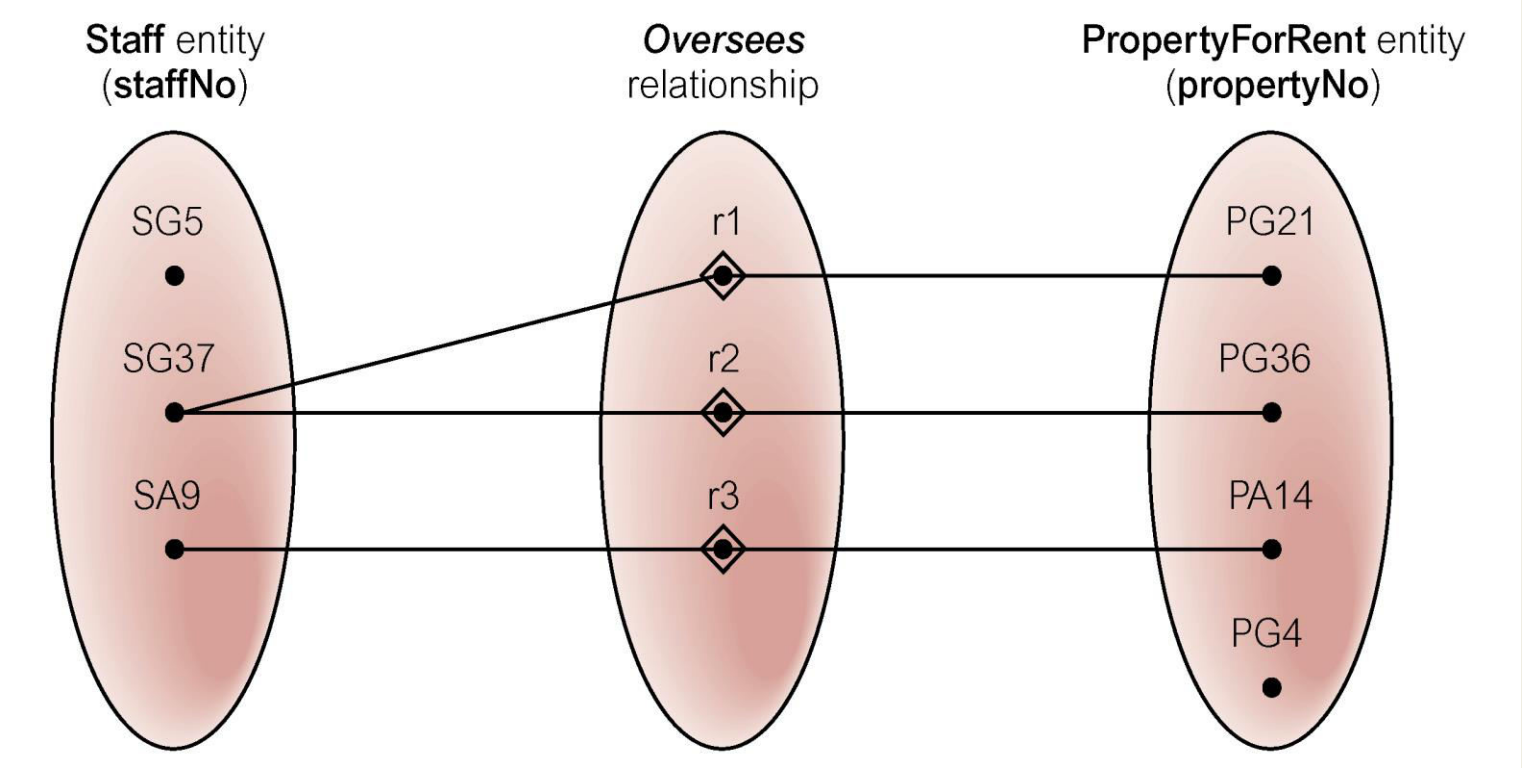
# Semantic net of Staff *Manages* Branch relationship type



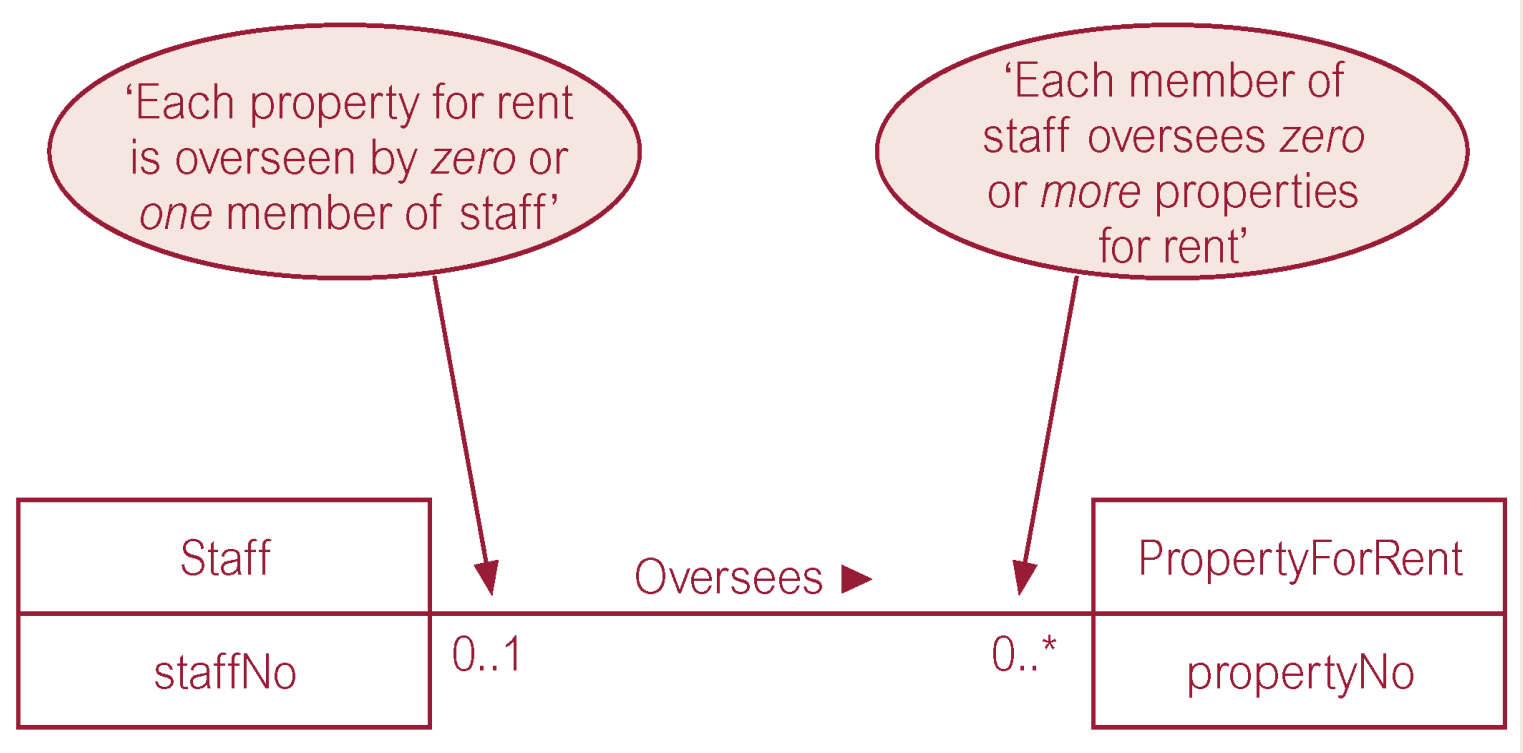
# Multiplicity of Staff *Manages* Branch (1:1) relationship



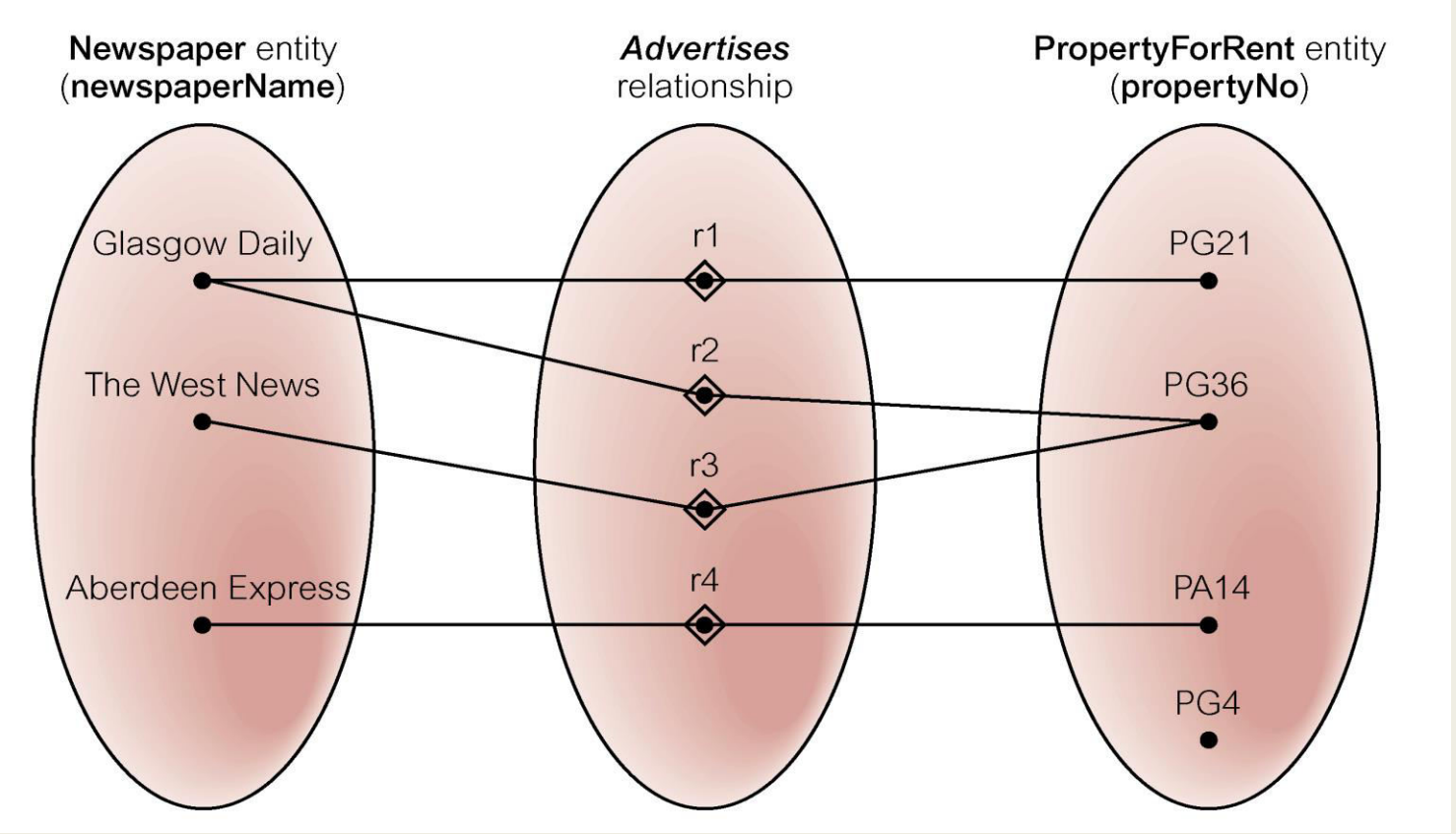
# Semantic net of Staff *Oversees* PropertyForRent relationship type



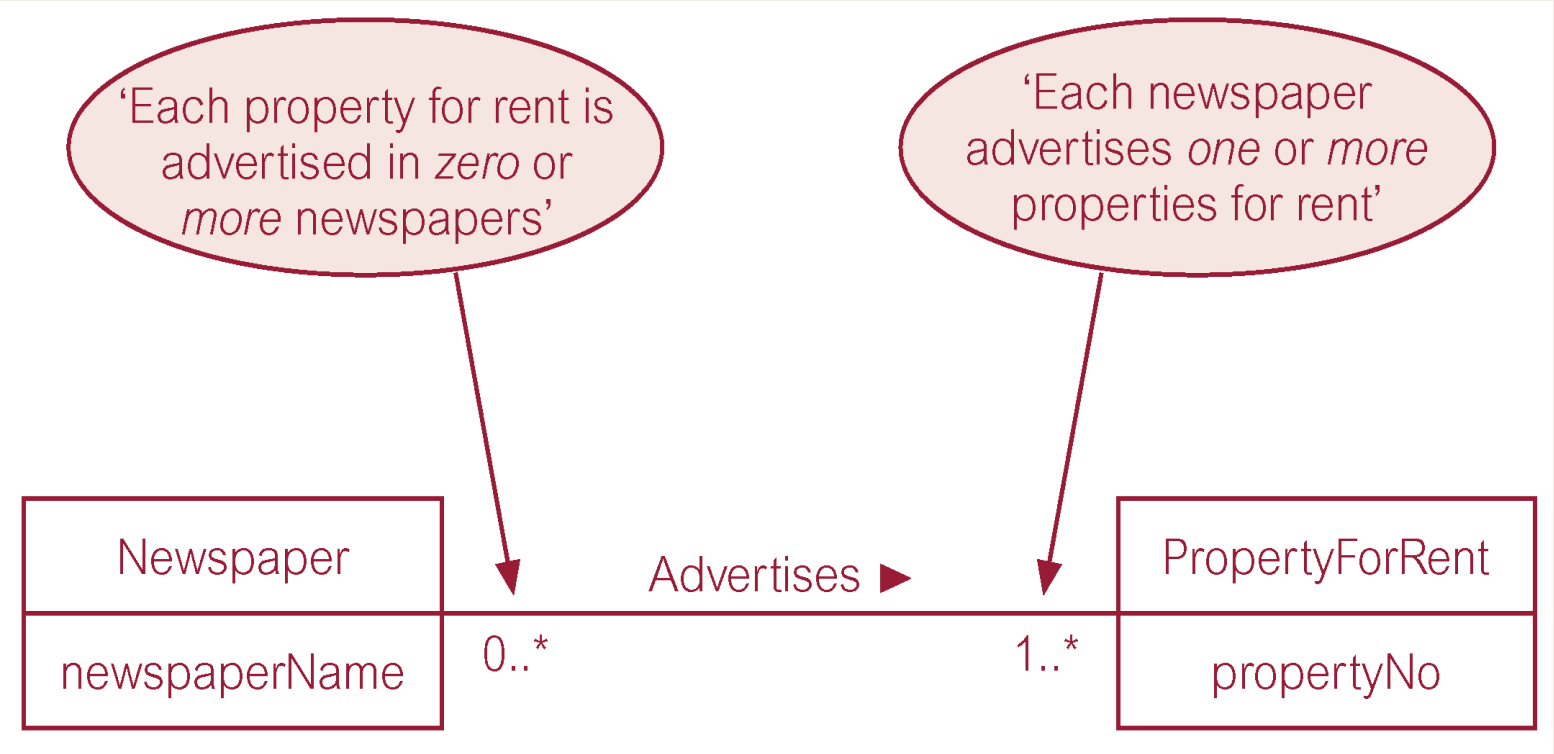
# Multiplicity of Staff *Oversees* PropertyForRent (1:*) relationship type



# Semantic net of Newspaper *Advertises* PropertyForRent relationship type



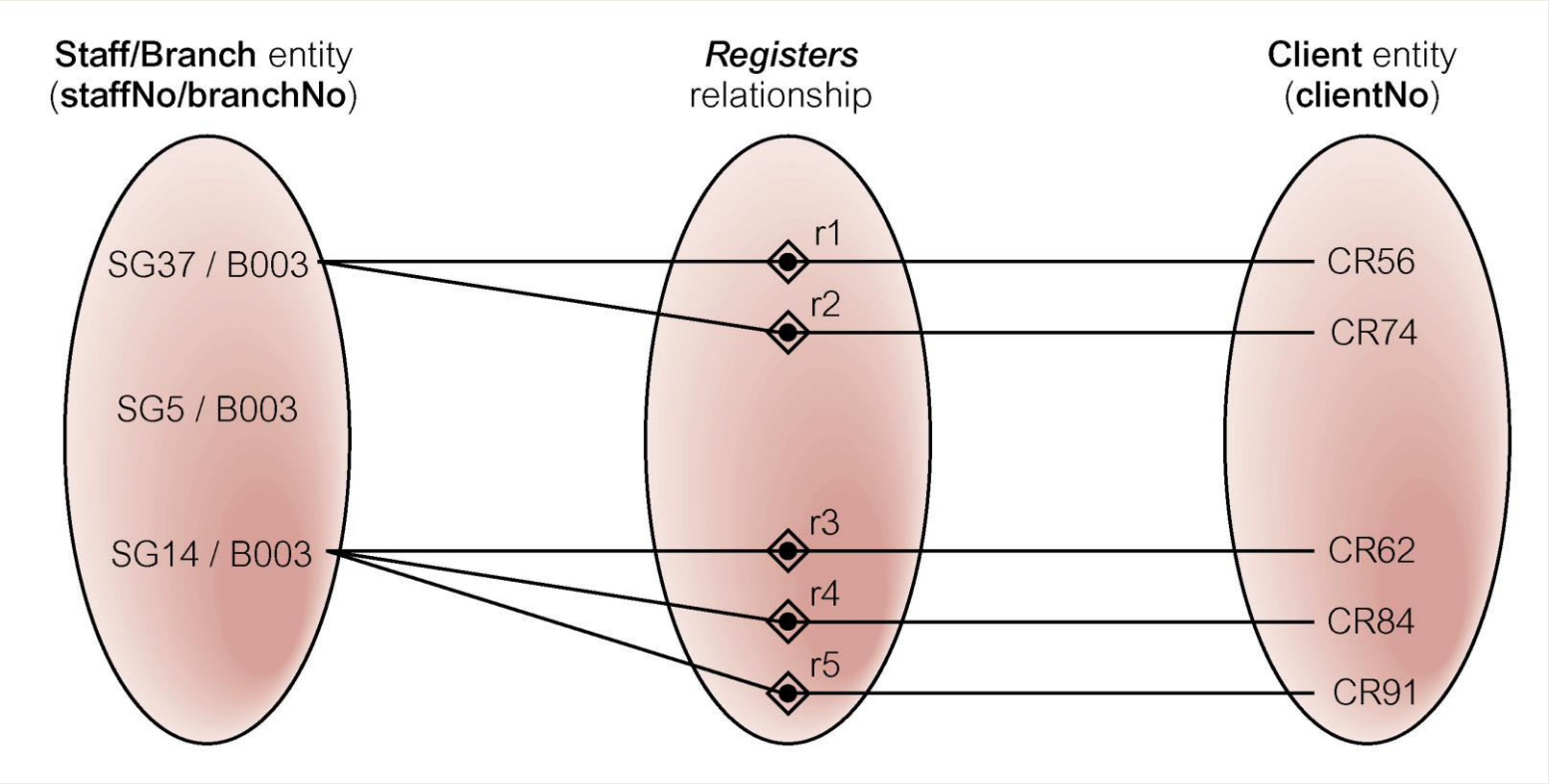
# Multiplicity of Newspaper *Advertises* PropertyForRent (*:*) relationship



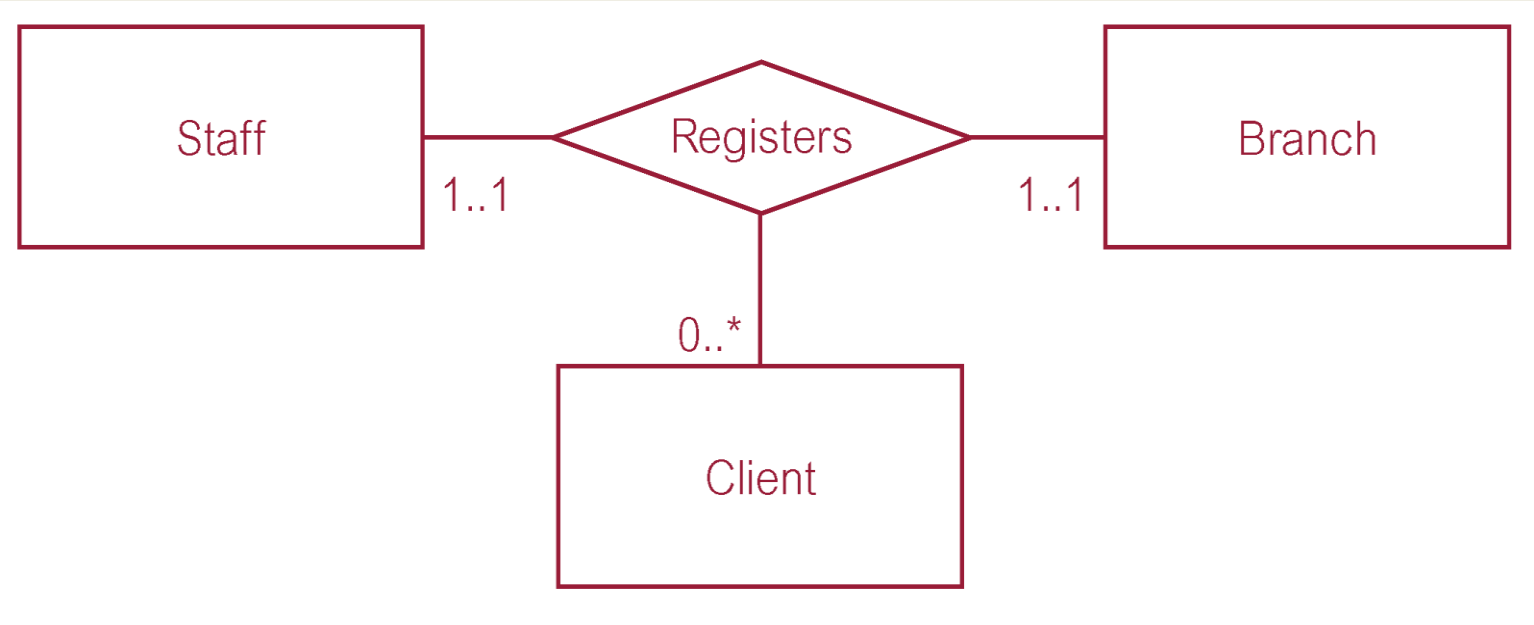
# Structural Constraints

- 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*.**

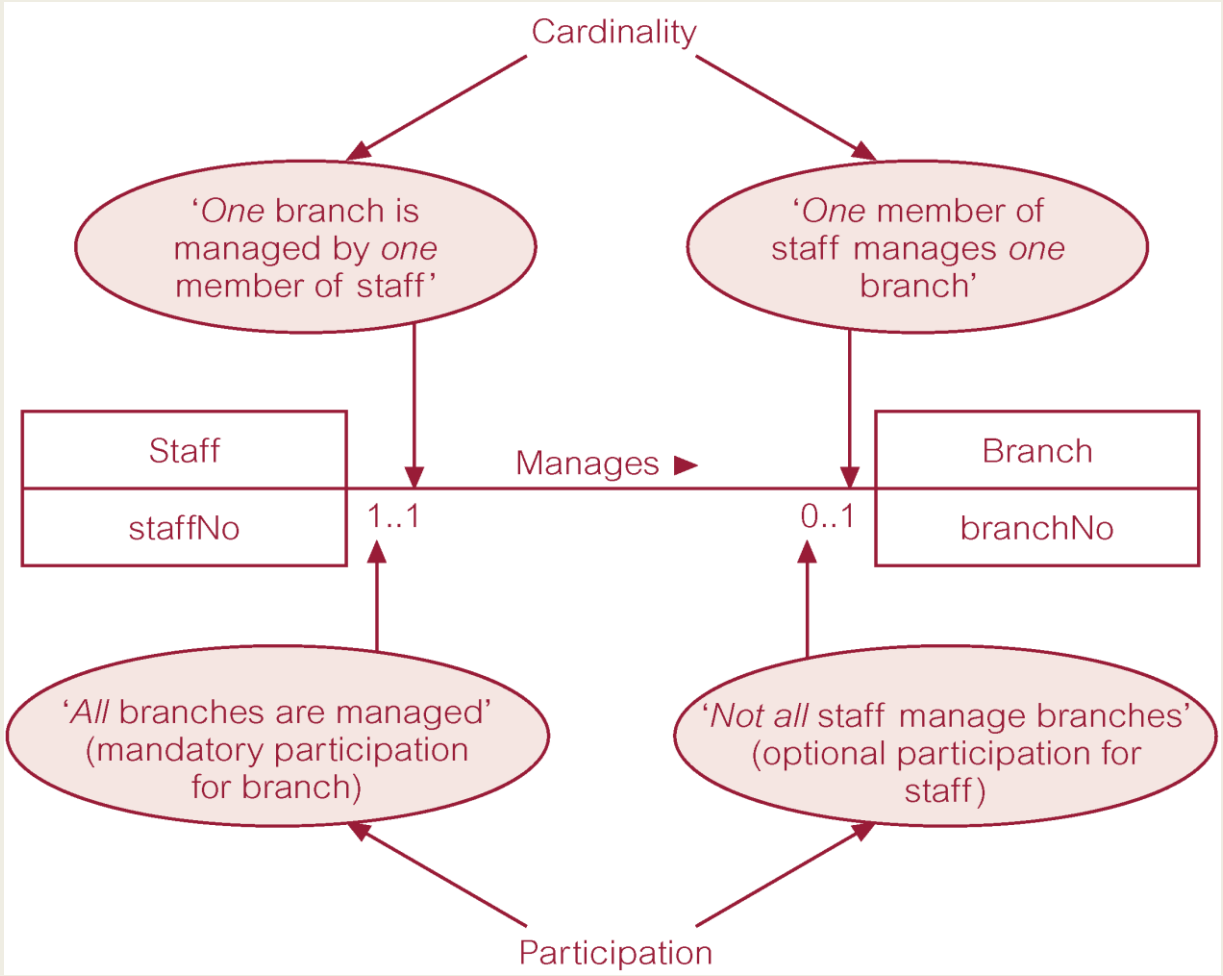
**Cardinality**

- 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.

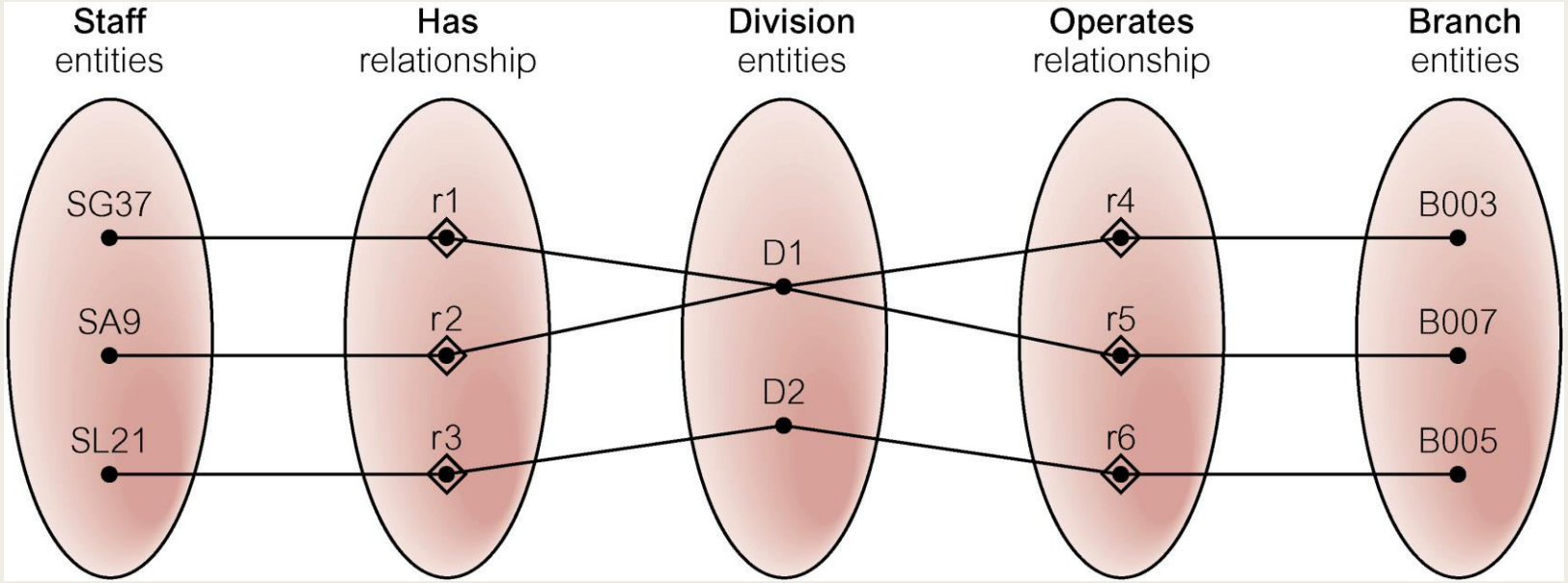
# Multiplicity as cardinality and participation constraints



# Problems with ER Modeling

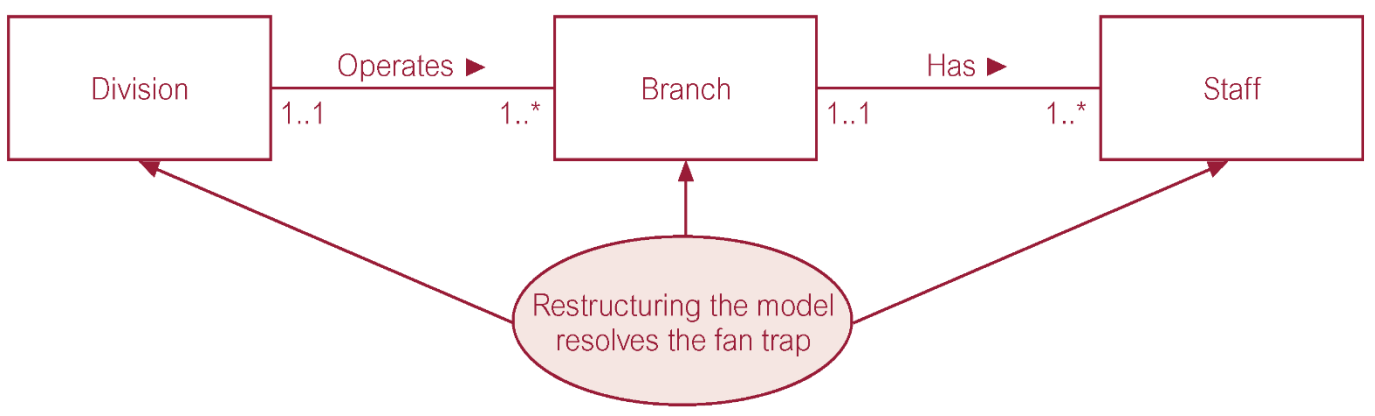
- Problems may arise when designing a conceptual data model called connection traps.
- Often due to a misinterpretation of the meaning of certain relationships.
- Fan Traps: Where a model represents a relationship between entity types, but pathway between certain entity occurrences is ambiguous.
- Chasm Trap: Where a model suggests the existence of a relationship between entity types, but pathway does not exist between certain entity occurrences.

# Fan Trap

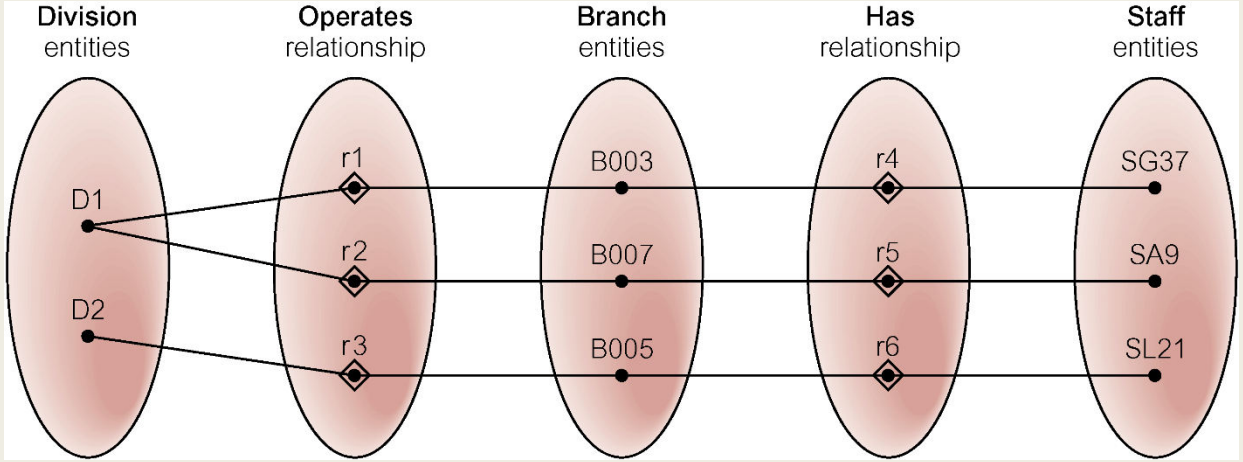




At which branch office does staff number SG37 work?

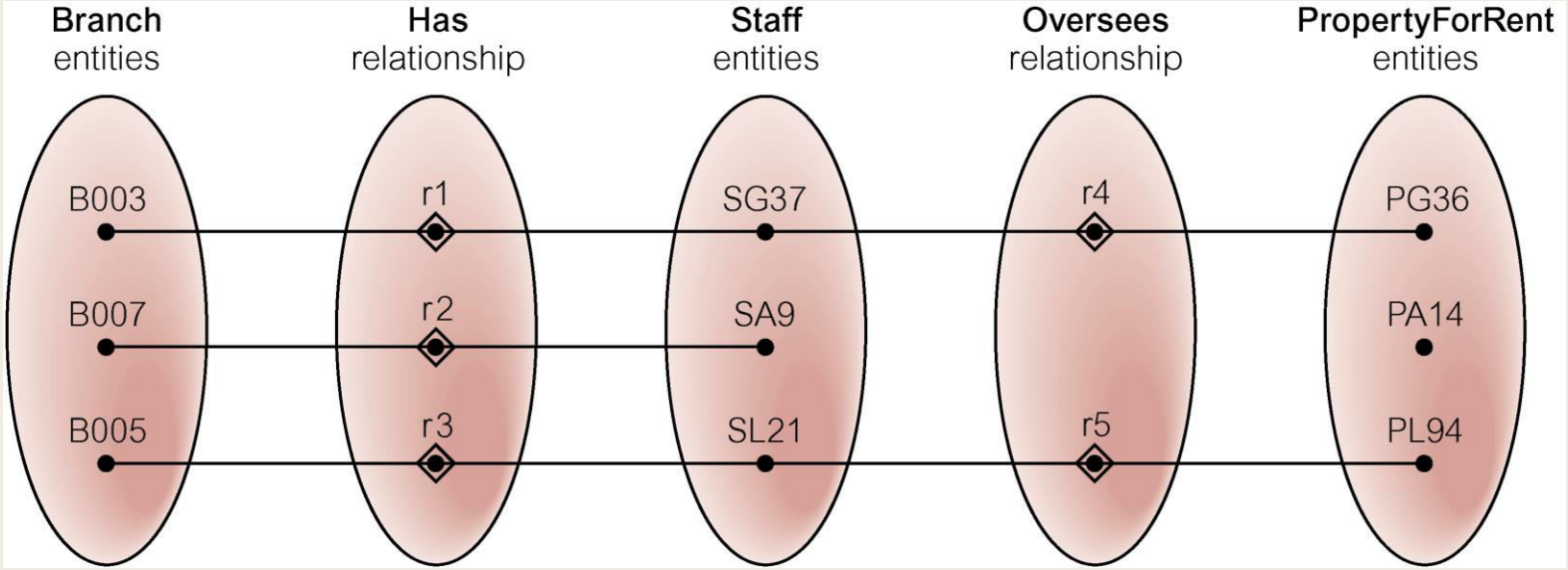
# Solution of Fan Trap




SG37 works at branch B003.

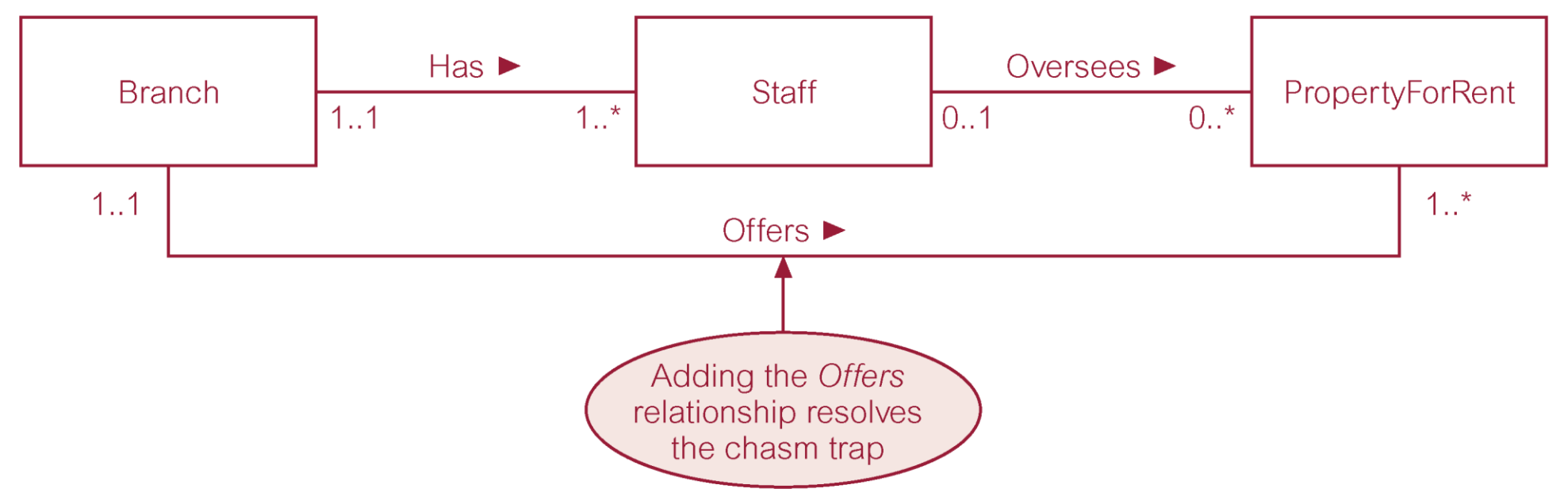


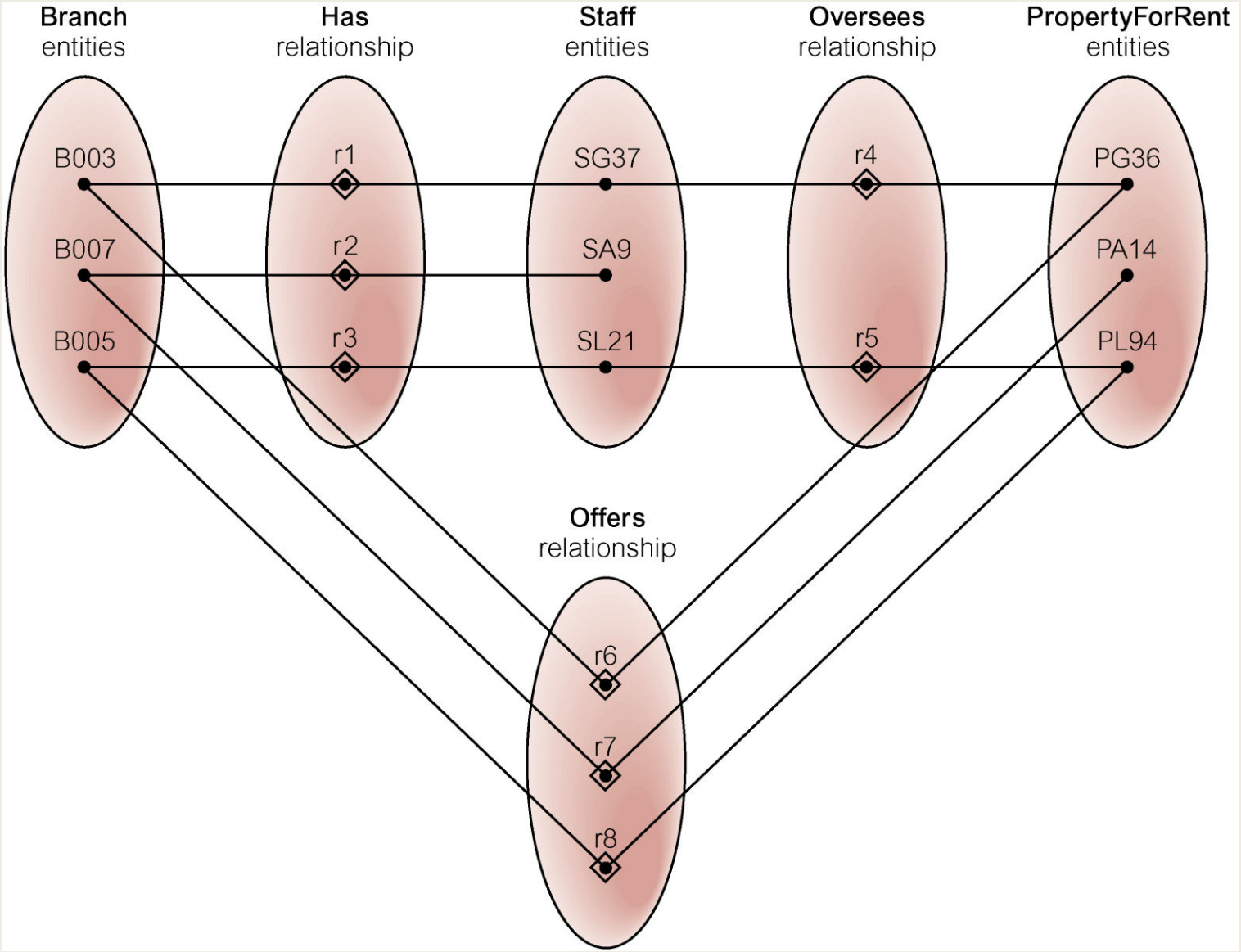
# Chasm Trap


At which branch office is property PA14 available?

# Solution of Chasm Trap




Branch
entities
Has
relationship
Staff
entities
Oversees
relationship
PropertyForRent
entities
B003
B007
B005
r1
r2
r3
SG37
SA9
SL21
r4
r5
PG36
PA14
PL94
Offers
relationship
r6
r7
r8

# The Enhanced Entity-Relationship Model

- Since 1980s there has been an increase in emergence of new database applications with more demanding requirements.
- Basic concepts of ER modeling are not sufficient to represent requirements of newer, more complex applications.
- Response is development of additional ‘semantic’ modeling concepts.
- Semantic concepts are incorporated into the original ER model and called the Enhanced Entity-Relationship (EER) model.
- Examples of additional concept of EER model is called specialization / generalization.

# Specialization / Generalization

**Superclass**

- An entity type that includes one or more distinct subgroupings of its occurrences.

**Subclass**

- A distinct subgrouping of occurrences of an entity type.

- Superclass/subclass relationship is one-to-one (1:1).
- Superclass may contain overlapping or distinct subclasses.
- NOT ALL members of a superclass need be a member of a subclass.

# Specialization / Generalization

**Attribute Inheritance**

- An entity in a subclass represents same ‘real world’ object as in superclass, and may possess subclass-specific attributes, as well as those associated with the superclass.

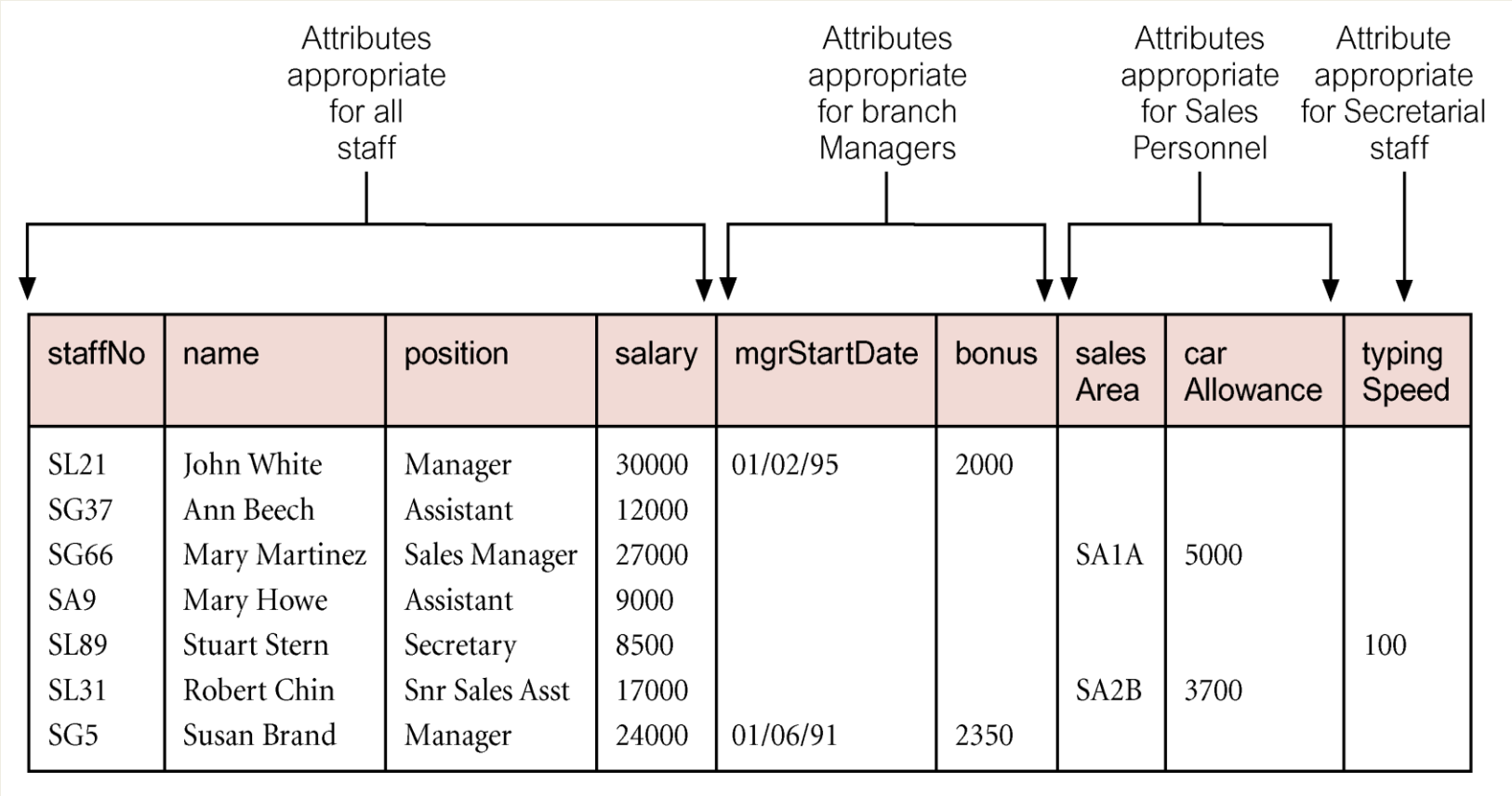
**Specialization**

- Process of maximizing differences between members of an entity by identifying their distinguishing characteristics.

**Generalization**

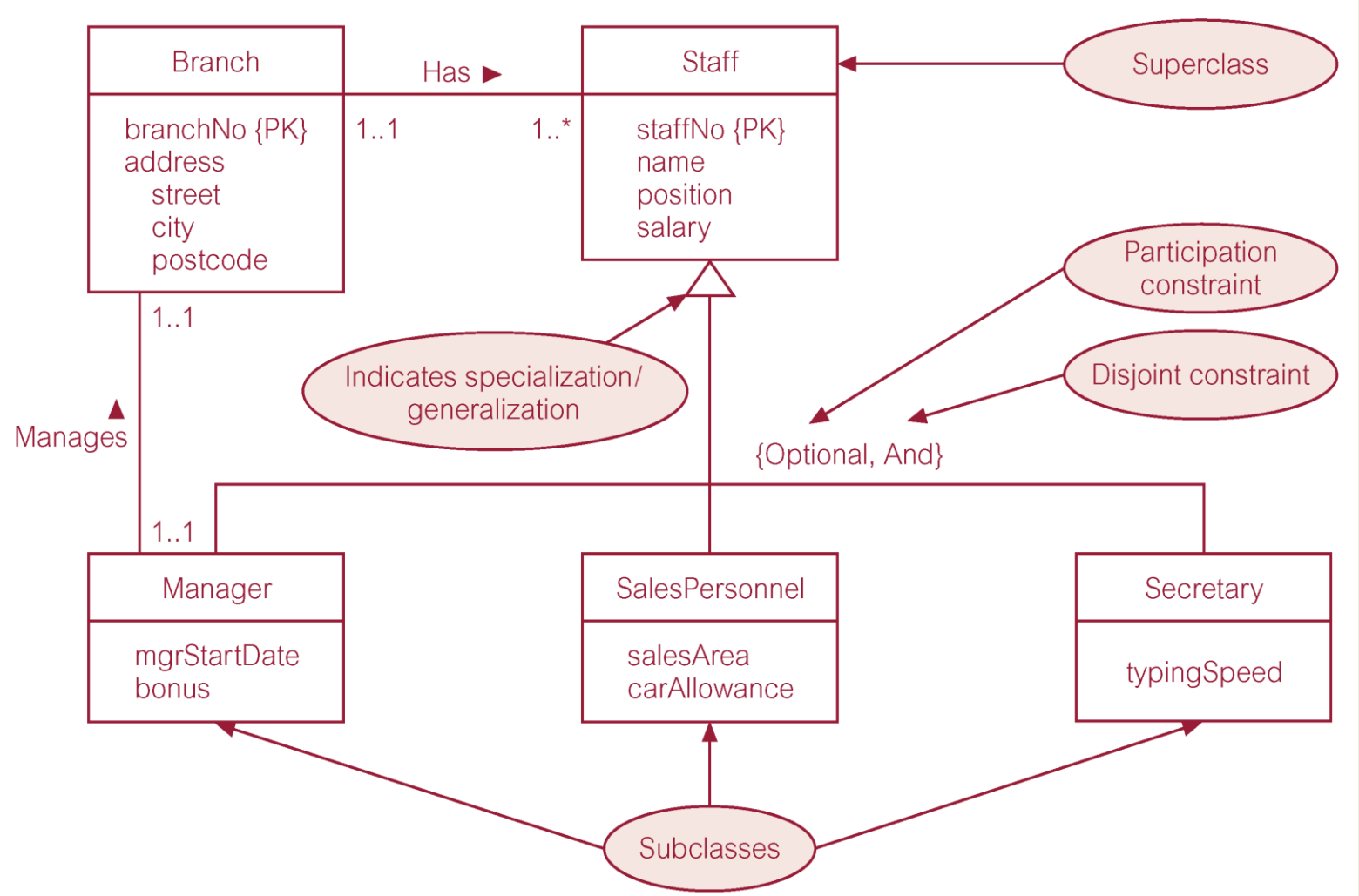
- Process of minimizing differences between entities by identifying their common characteristics.

# AllStaff relation holding details of all staff

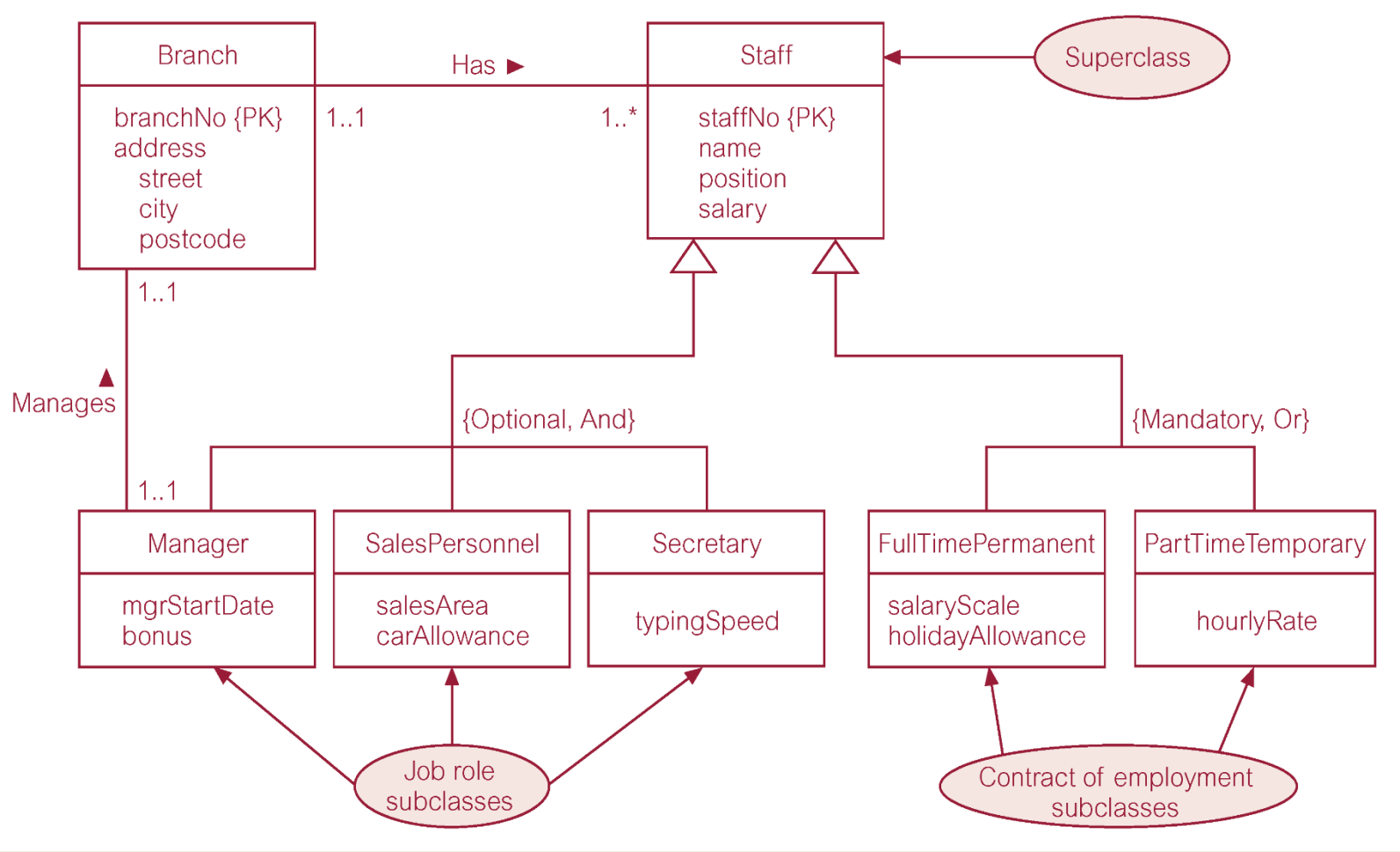



| staffNo | name | position | salary | mgrStartDate | bonus | sales Area | car Allowance | typing Speed |
|---|---|---|---|---|---|---|---|---|
| SL21 | John White | Manager | 30000 | 01/02/95 | 2000 | | | |
| SG37 | Ann Beech | Assistant | 12000 | | | | | |
| SG66 | Mary Martinez | Sales Manager | 27000 | | | SA1A | 5000 | |
| SA9 | Mary Howe | Assistant | 9000 | | | | | |
| SL89 | Stuart Stern | Secretary | 8500 | | | | | 100 |
| SL31 | Robert Chin | Snr Sales Asst | 17000 | | | SA2B | 3700 | |
| SG5 | Susan Brand | Manager | 24000 | 01/06/91 | 2350 | | | |

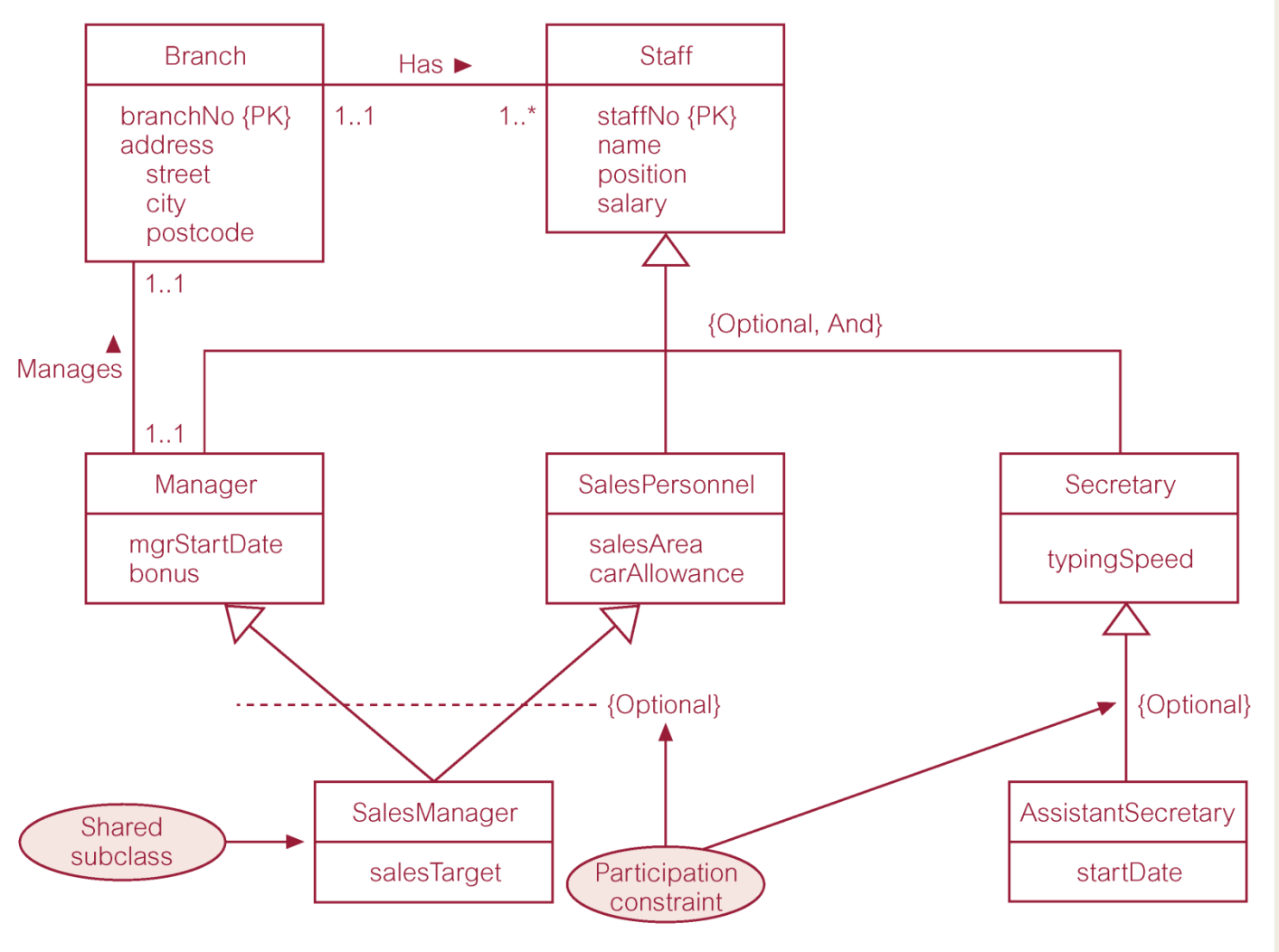
# Specialization/generalization of Staff entity into subclasses representing job roles



# Specialization/generalization of Staff entity into job roles and contracts of employment



# EER diagram with shared subclass and subclass with its own subclass



# Constraints on Specialization / Generalization

**Participation constraint**

- Determines whether every member in superclass must participate as a member of a subclass.
- May be ***mandatory*** or ***optional***.

**Disjoint constraint**

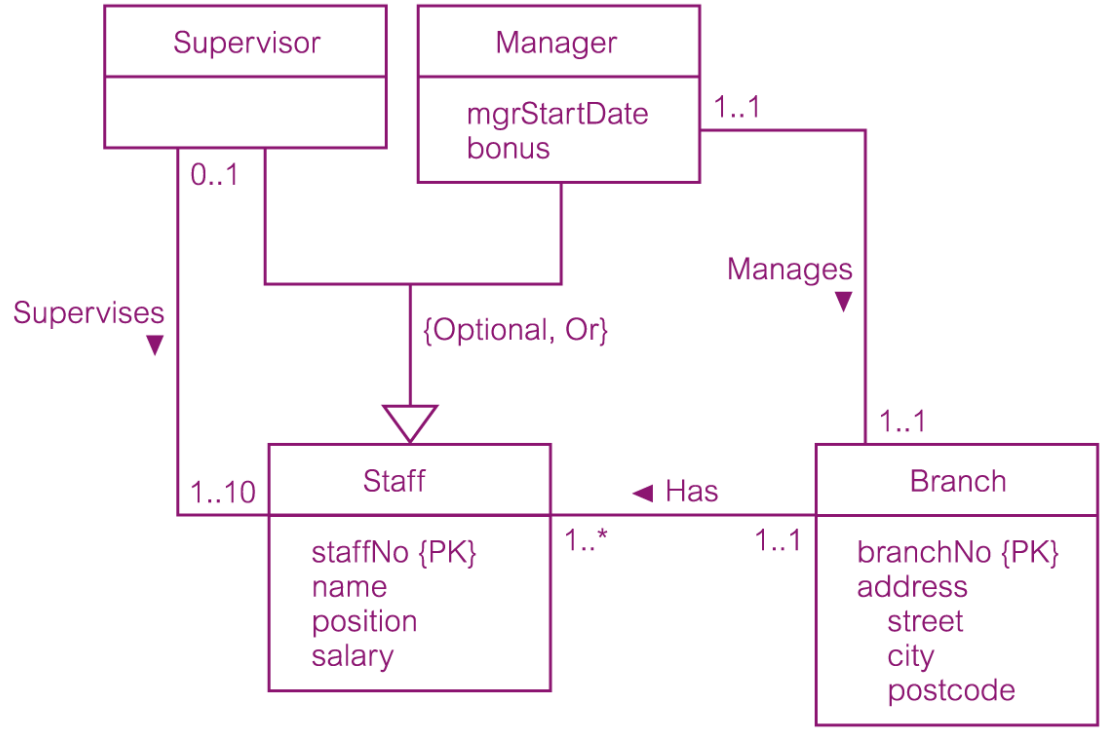
- Describes relationship between members of the subclasses and indicates whether member of a superclass can be a member of one, or more than one, subclass.
- May be ***disjoint*** or ***nondisjoint***.

# Constraints on Specialization / Generalization

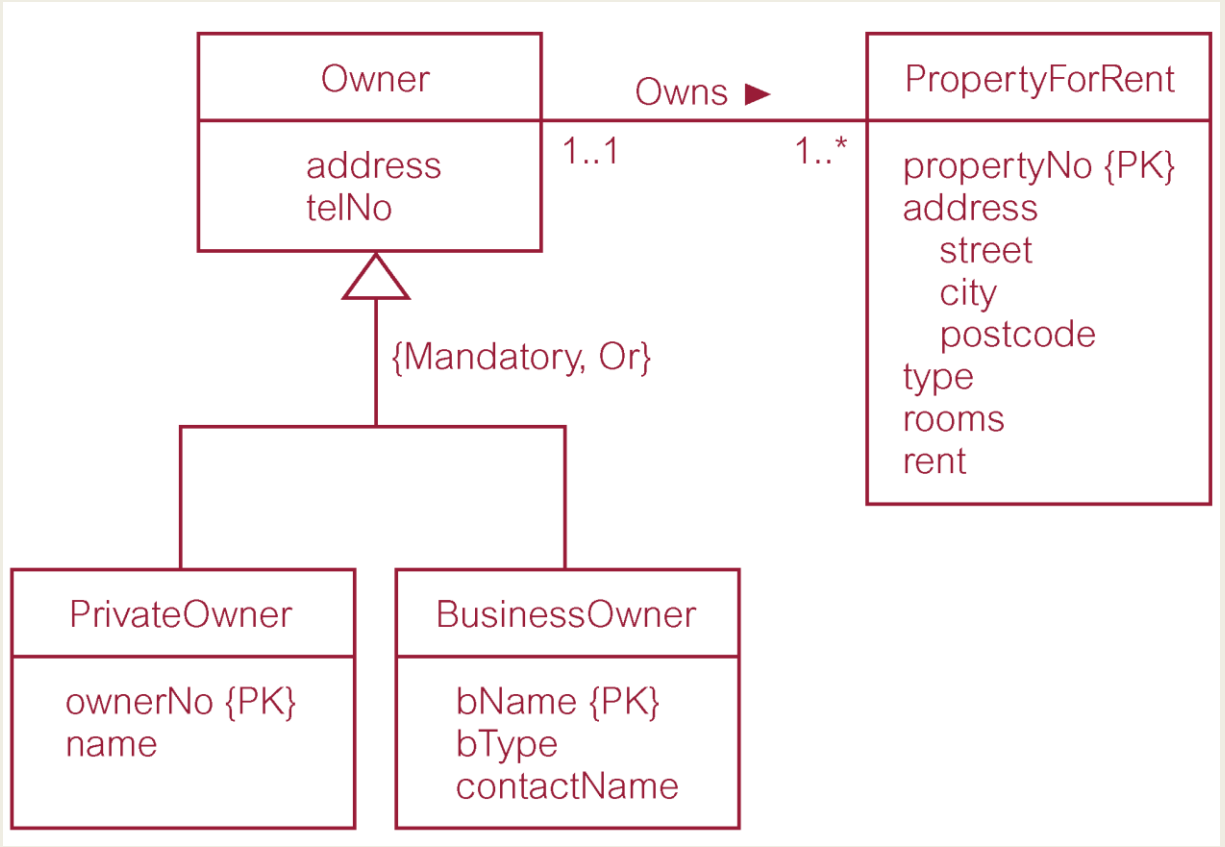
- There are four categories of constraints of specialization and generalization:
  - *mandatory and disjoint (or)*
  - *optional and disjoint*
  - *mandatory and nondisjoint (and) → overlap*
  - *optional and nondisjoint.*

# *DreamHome* worked example - Staff Superclass with Supervisor and Manager subclasses

**Figure 12.5**
Staff superclass with Supervisor and Manager subclasses.



# *DreamHome* worked example - Owner Superclass with PrivateOwner and BusinessOwner subclasses



# *DreamHome* worked example - Person superclass with Staff, PrivateOwner, and Client subclasses

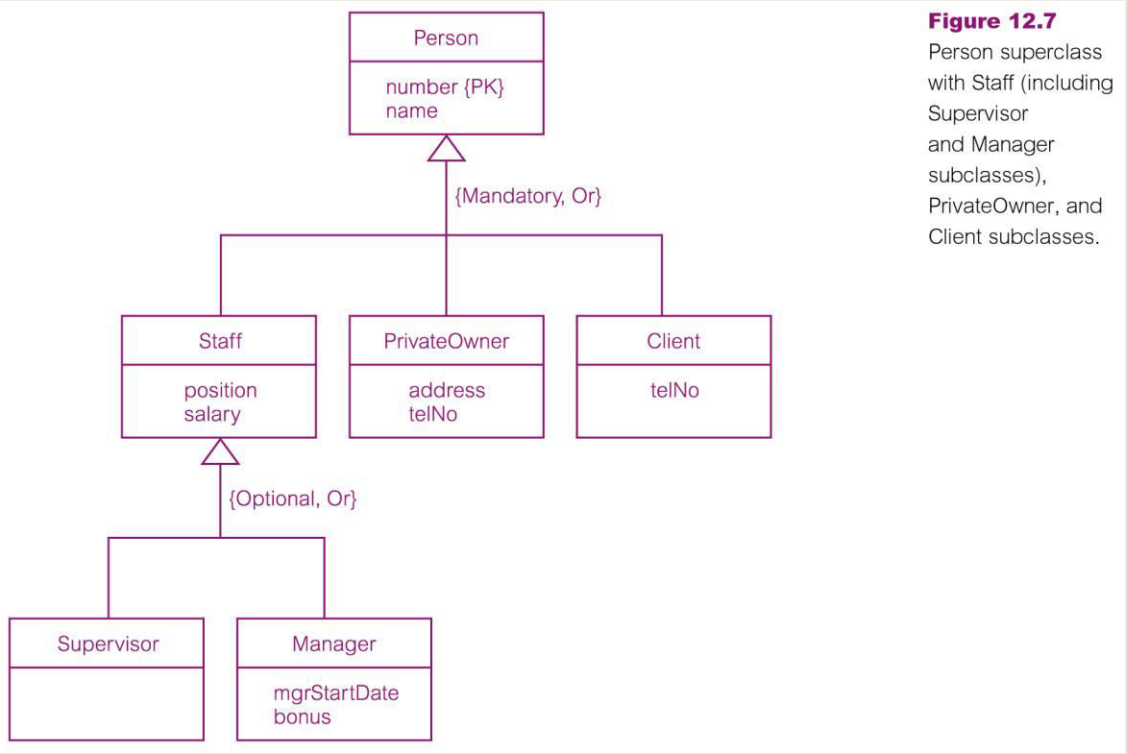



**Figure 12.7**
Person superclass with Staff (including Supervisor and Manager subclasses), PrivateOwner, and Client subclasses.

# References

*Database Systems: A Practical Approach to Design, Implementation, and Management,* Thomas Connolly and Carolyn Begg, 5th Edition, 2010, Pearson.