

# Chapter 11

---

## Entity-Relationship Modelling

# ER modeling

- **Top-down approach to database design**
- **Start by identifying the important data (called entities) and relationships between the data**

# Entity and Entity Types

- **Anything about which we want to collect and store the data is called an Entity**
- **A group of objects with the same properties, which are identified by the enterprise as having an independent existence**

# Entities with physical and conceptual existence

## Physical existence

Staff

Part

Property

Supplier

Customer

Product

## Conceptual existence

Viewing

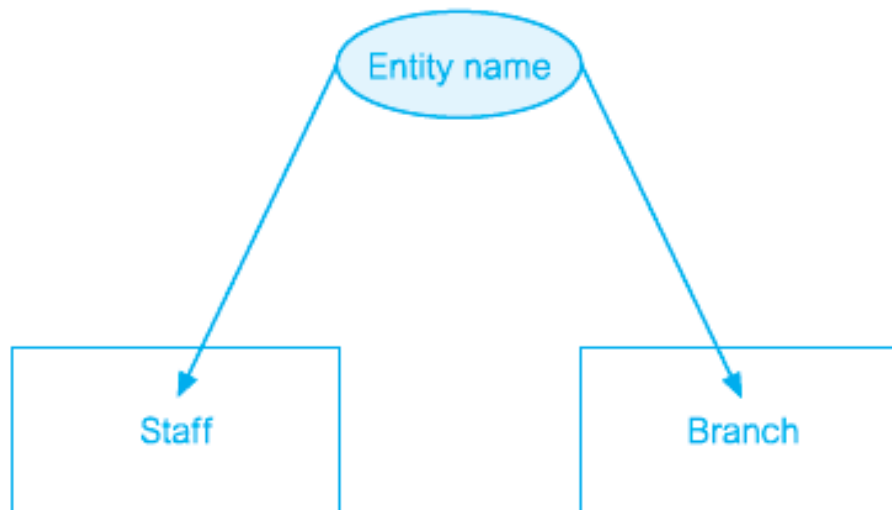
Sale

Inspection

Work experience

# Diagrammatic Representation of Entity Types

- Each entity type is shown as a rectangle labeled with the name of the entity, which is normally a singular noun
- In UML, the first letter of each word in the entity name is upper case (for example, Staff and PropertyForRent)



# Entity Occurrence

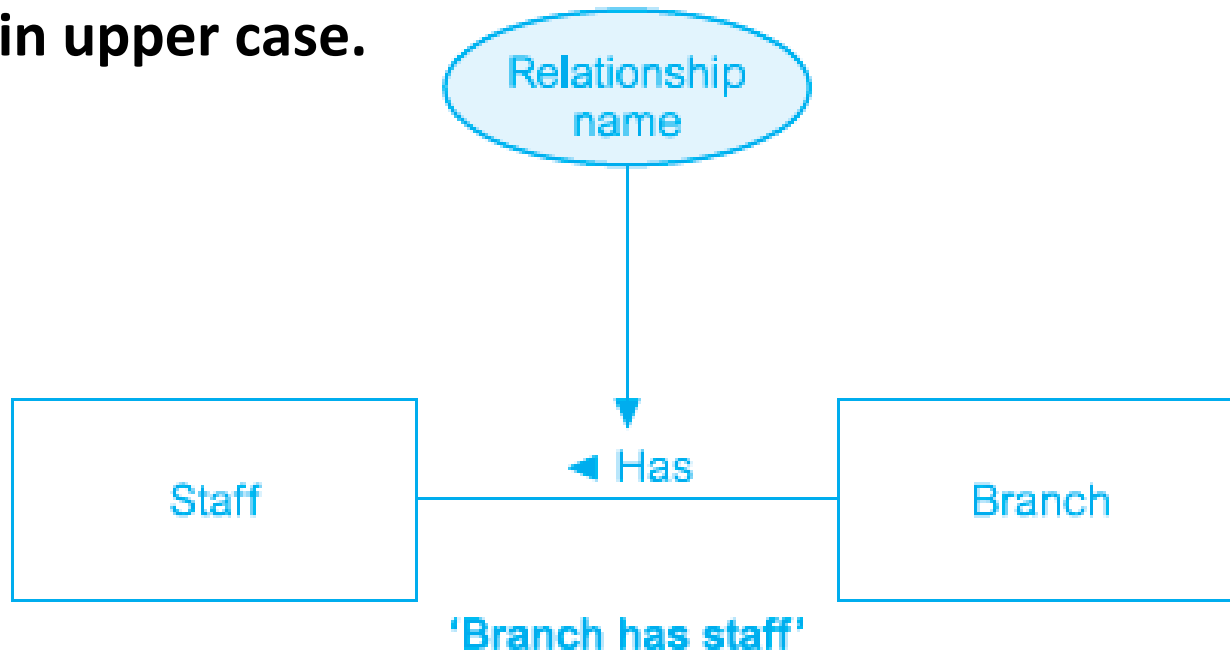
- A uniquely identifiable object of an entity type
- We represent each Branch and Staff entity occurrences using values for their primary key attributes E.g., branchNo and staffNo

# Relationships

- **Relationships Types**
  - A set of meaningful associations among entity Types
- **Relationship Occurrence**
  - A uniquely identifiable association, which includes one occurrence from each participating entity type

# Diagrammatic Representation of Relationship Types

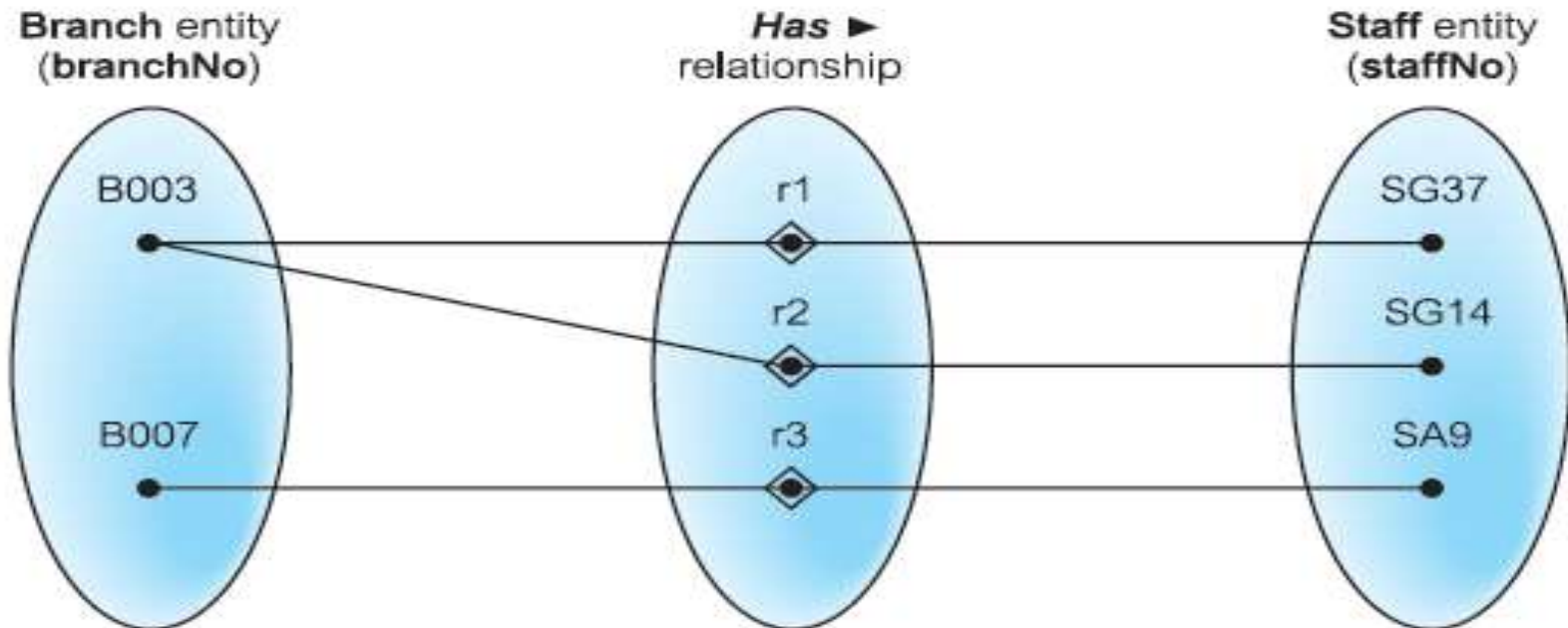
- Each relationship type is shown as a line connecting the associated entity types, labeled with the name of the relationship
- Normally, a relationship is named using a verb (for example, *Supervises* or *Manages*) or a short phrase including a verb (for example, *LeasedBy*)
- Again, the first letter of each word in the relationship name is shown in upper case.





# Diagrammatic Representation of Relationship Occurrence

- Relationship occurrence r1 represents the association between Branch entity B003 and Staff entity SG37.



# Degree of a Relationship Type

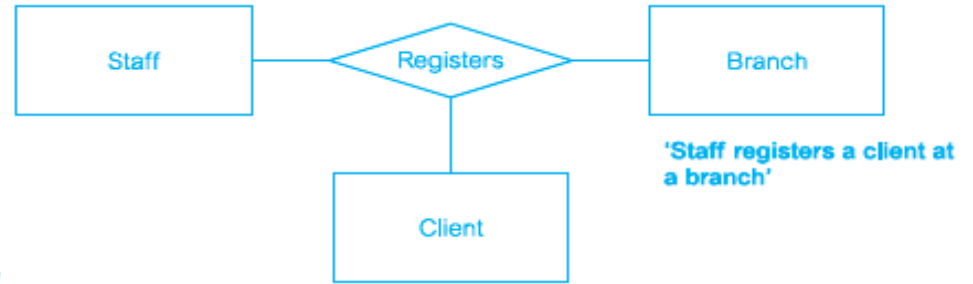
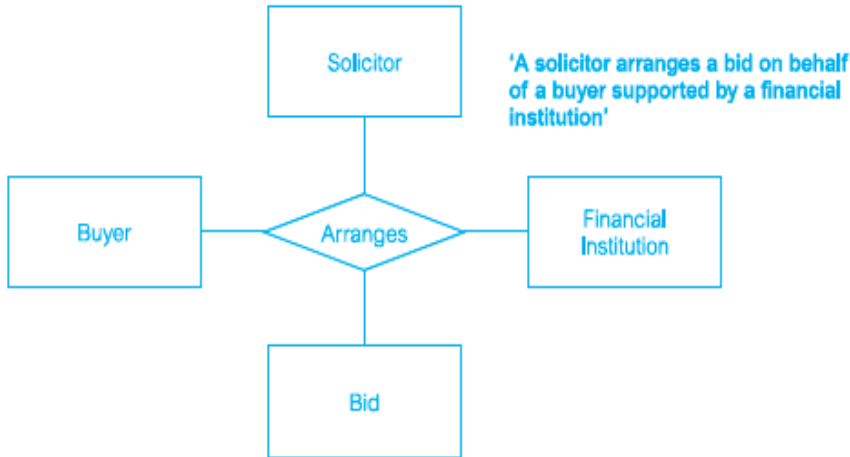
- Number of participating entities in relationship
- The UML notation uses a diamond to represent relationships with degrees higher than binary

'Private owner owns property for rent'

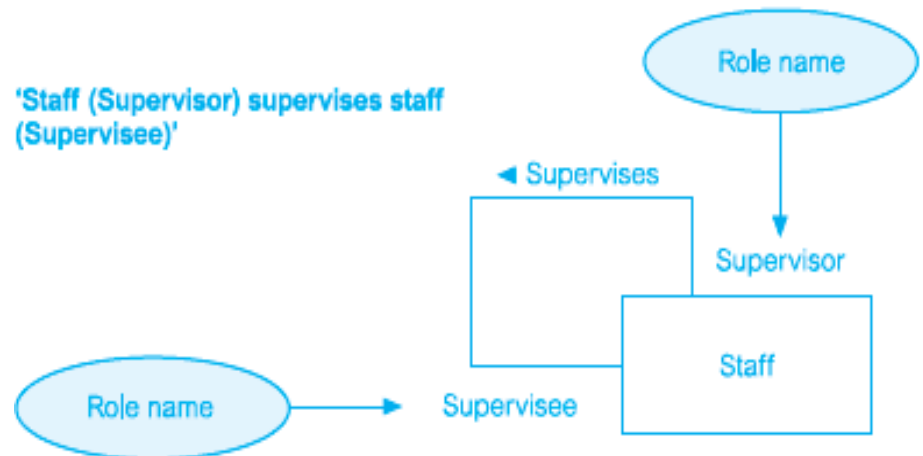


**Binary Relationship**

**Ternary Relationship**



**Quaternary Relationship**



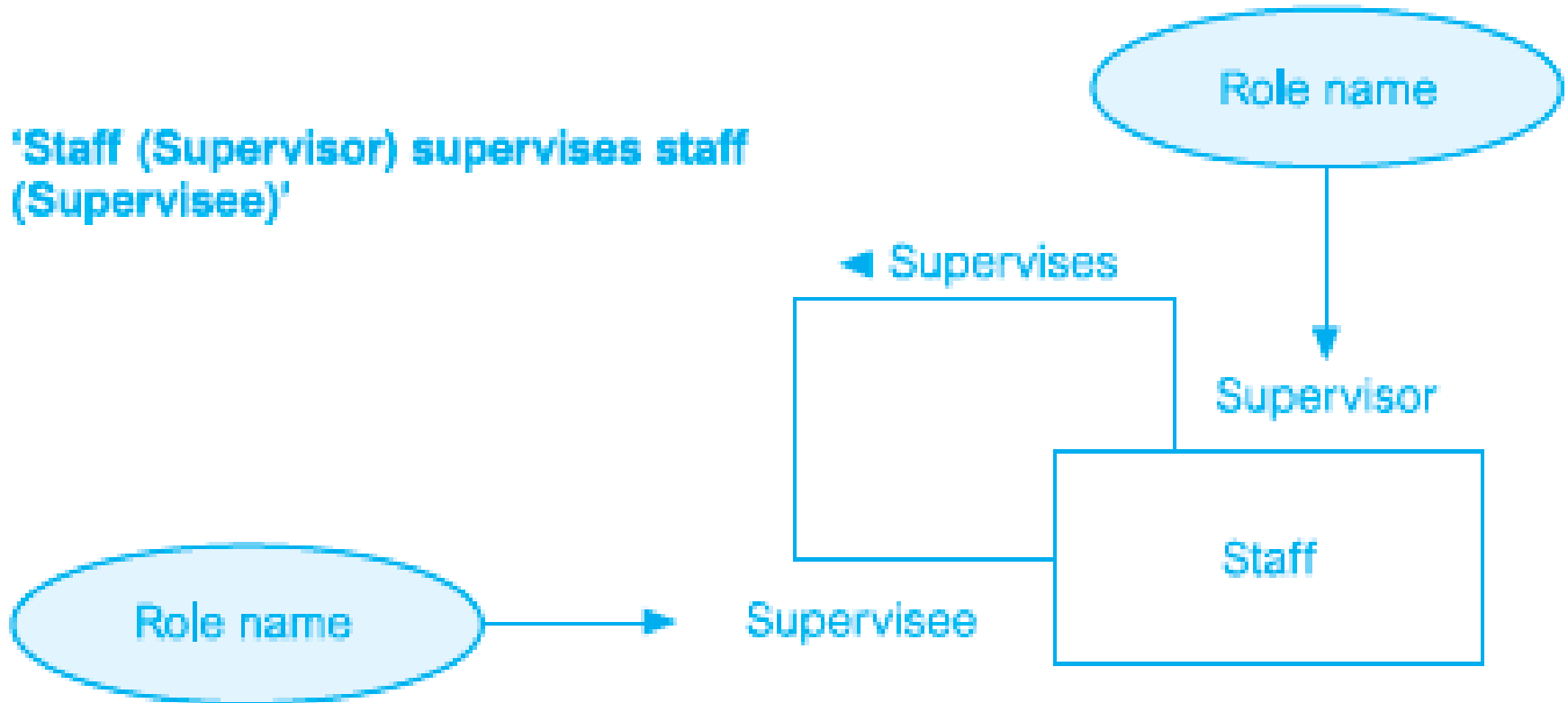
**Unary Relationship**

# Recursive relationships

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

# An example of a recursive relationship called *Supervises* with role names Supervisor and Supervisee

'Staff (Supervisor) supervises staff (Supervisee)'



# Attributes

- **Property of an entity or a relationship**
- **Represent the main source of data stored in the database**

# Attributes

- **Attribute can be classified as being:**
  - simple or composite
  - single-valued or multi-valued
  - or derived

# Simple Attribute

- Attribute composed of a single component
- Simple attributes cannot be further subdivided into smaller components
- Examples of simple attributes include position and salary of the Staff entity
- Simple attributes are sometimes called *atomic* attributes



# Composite Attribute

- Attribute composed of multiple components
- Some attributes can be further divided to yield smaller components with an independent existence of their own
- For example, the address attribute of the Branch entity with the value (163 Main St, Glasgow, G11 9QX) can be subdivided into street (163 Main St), city (Glasgow), and postcode (G11 9QX) attributes

# Single-Valued Attribute

- **Attribute that holds a single value for an entity occurrence**
- **The majority of attributes are single-valued**
- **For example, each occurrence of the branchNo attribute is single valued (for example B003), and therefore the branchNo attribute is referred to as being single-valued**

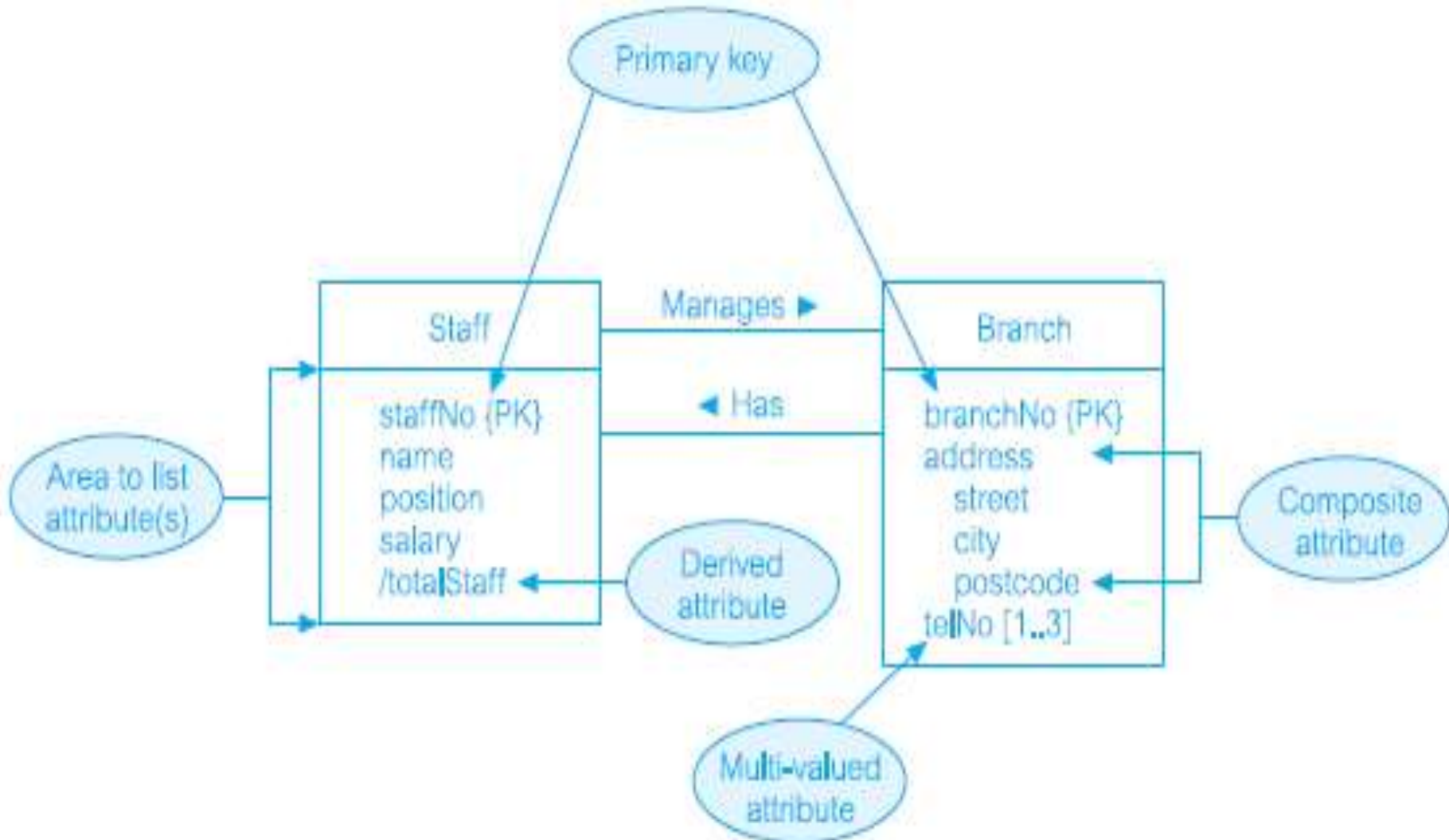
# Multi-valued attribute

- Attribute that holds multiple values for an entity occurrence
- For example, each occurrence of the Branch entity type can have multiple values for the telNo attribute (for example, branch number B003 has telephone numbers 0141-339-2178 and 0141-339-4439) and therefore the telNo attribute in this case is multi-valued
- A multi-valued attribute may have a set of numbers with upper and lower limits
- For example, the telNo attribute of the Branch entity type has between one and three values
- In other words, a branch may have a minimum of a single telephone number to a maximum of three telephone numbers

# 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

# ER diagram of entities and their attributes



# Keys

- **Superkey**
  - An attribute, or set of attributes, that uniquely identifies each entity occurrence
- **Candidate key**
  - A superkey that contains only the minimum number of attributes necessary for unique identification of each entity occurrence

# Keys

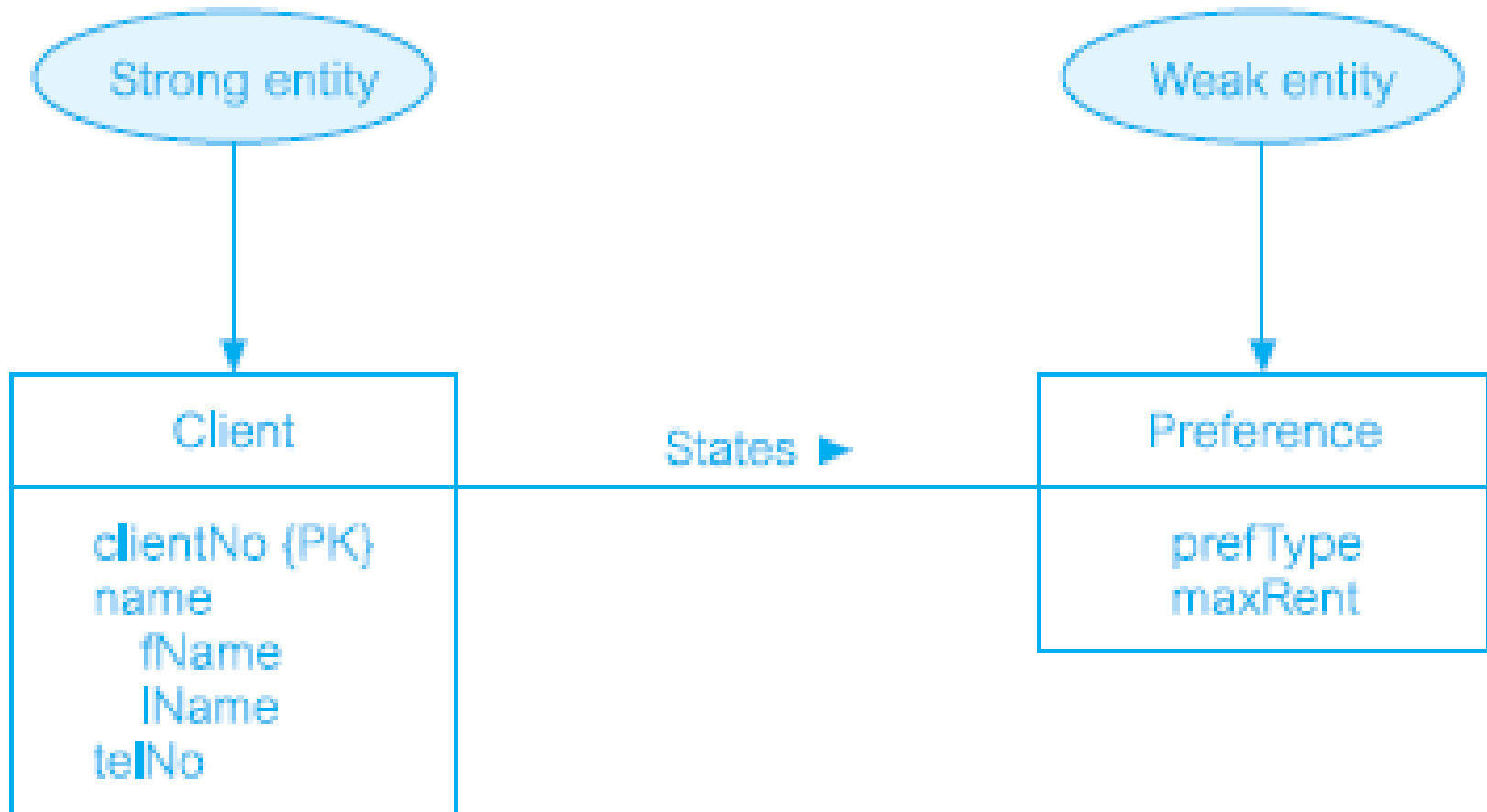
- **Primary key**
  - The candidate key that is selected to identify each entity occurrence
- **Alternate key**
  - The candidate keys that are not selected as the primary key of the entity

# Strong and weak entities

- **Strong entity**
  - An entity type that is *not existence-dependent on some other entity type*
- **Weak entity**
  - An entity type that is existence-dependent on some other entity type



# A strong entity type called Client and a weak entity type called Preference

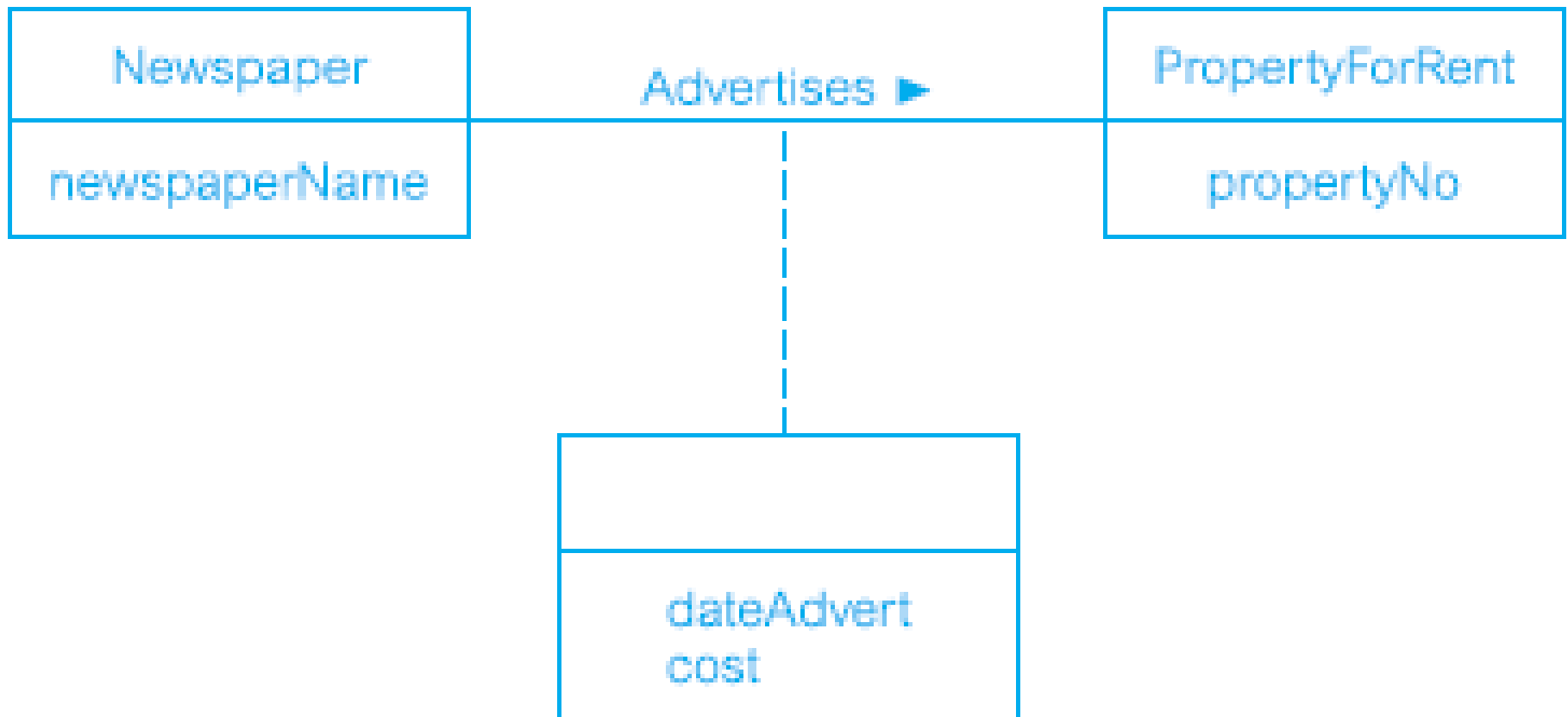


# Attributes on Relationships

- Attributes can also be assigned to relationships
- For example, consider the relationship *Advertises*, which associates the *Newspaper* and *PropertyForRent* entity types as shown in Figure
- To record the date the property was advertised and the cost, we associate this information with the *Advertises relationship* as attributes called *dateAdvert* and *cost*, rather than with the *Newspaper* or the *PropertyForRent* entities

An example of a relationship called *Advertises* with attributes dateAdvert and cost

'Newspaper advertises property for rent'



# Diagrammatic representation of attributes on relationships

- We represent attributes associated with a relationship type using the same symbol as an entity type
- However, to distinguish between a relationship with an attribute and an entity, the rectangle representing the attribute(s) is associated with the relationship using a dashed line
- For example, Figure shows the *Advertises* relationship with the attributes *dateAdvert* and *cost*

# Multiplicity Constraints on Relationships

- **Multiplicity constrains the way that entities are related**
- **The 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**

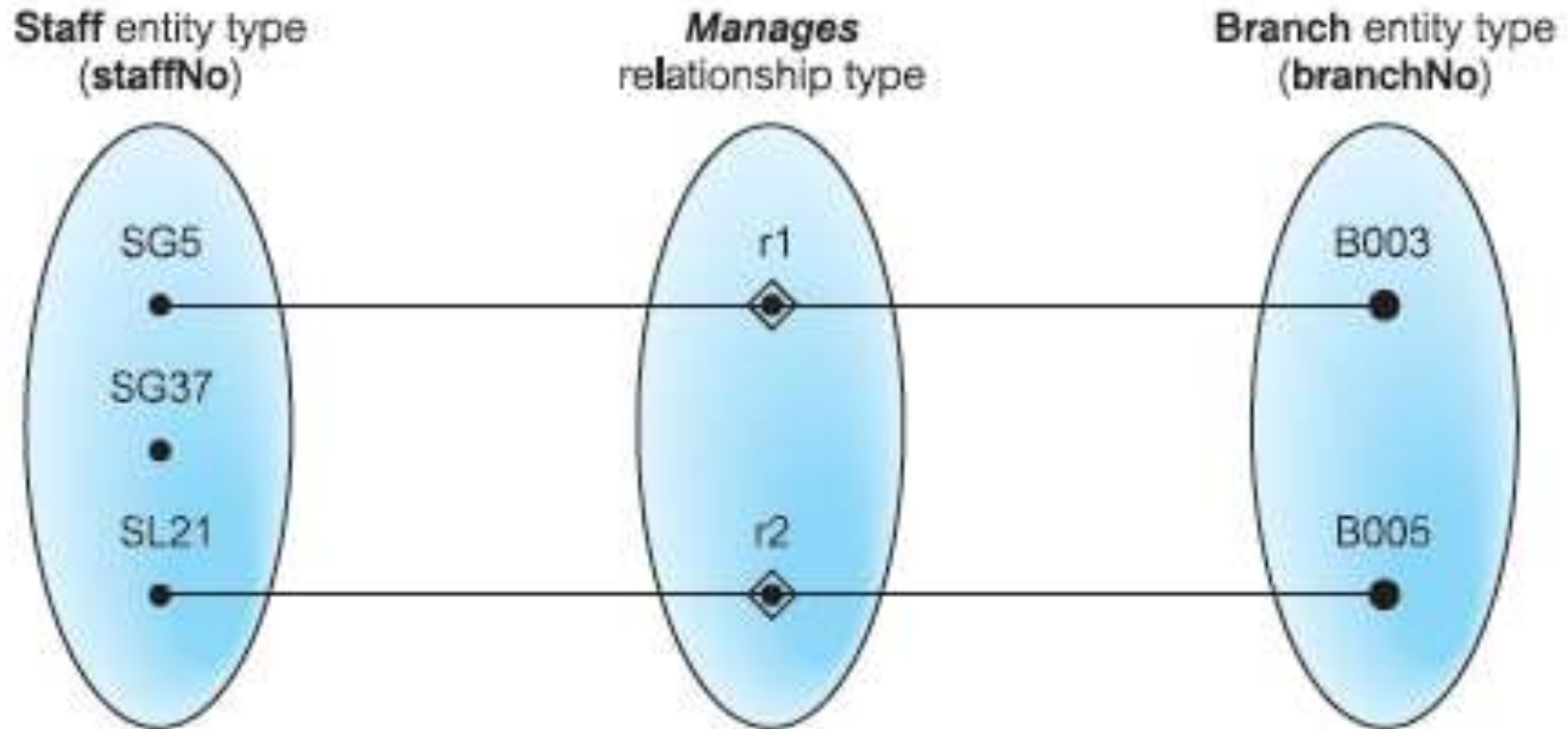
# Multiplicity constraints

- As discussed earlier, 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 (\*:\*)

# One-to-One (1:1) Relationships

- A member of staff manages a branch (1:1)
- Each relationship represents the association between a single Staff entity occurrence and a single Branch entity occurrence

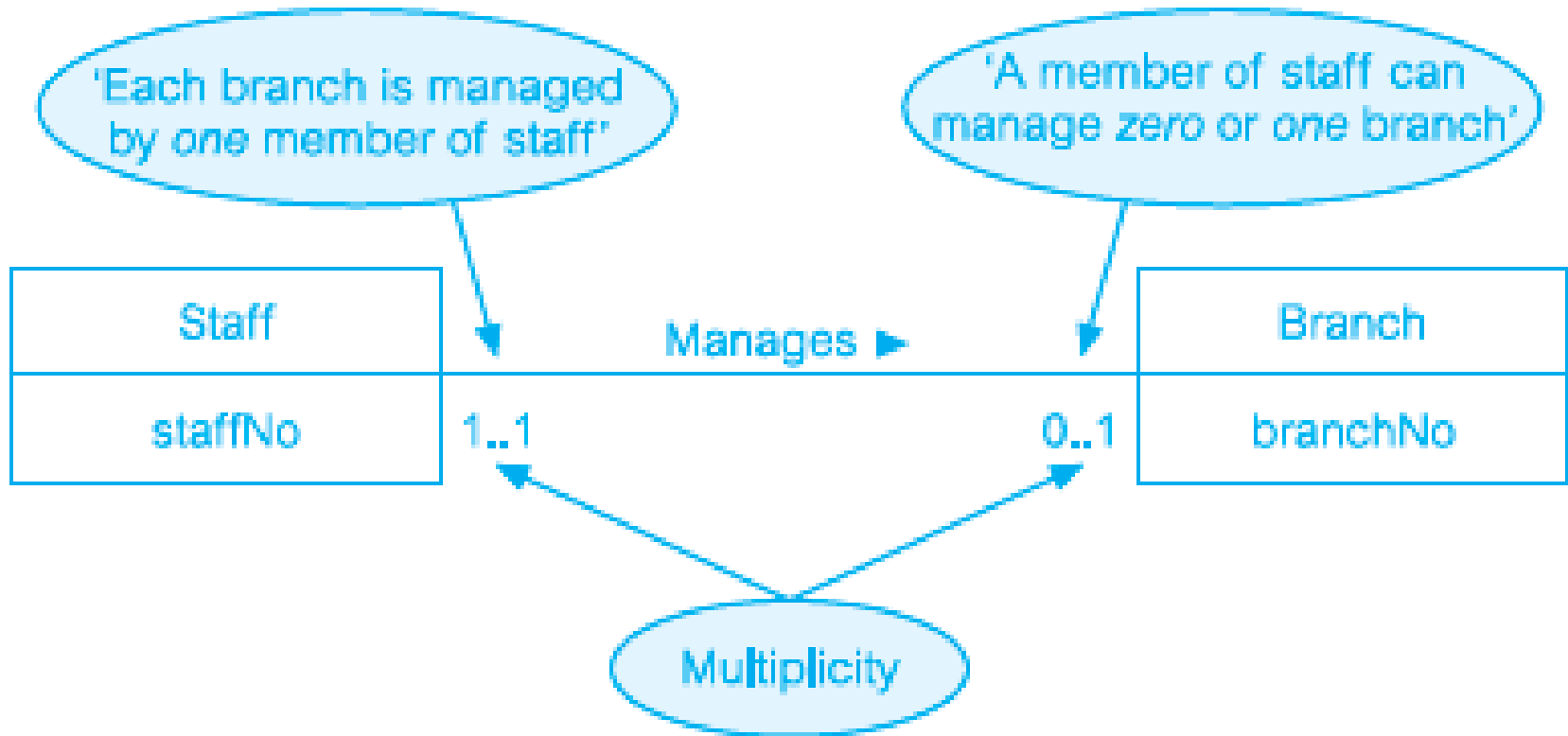
# Determining the multiplicity



Semantic net showing two occurrences of the Staff *Manages* Branch relationship type



# Diagrammatic representation of 1:1 relationships

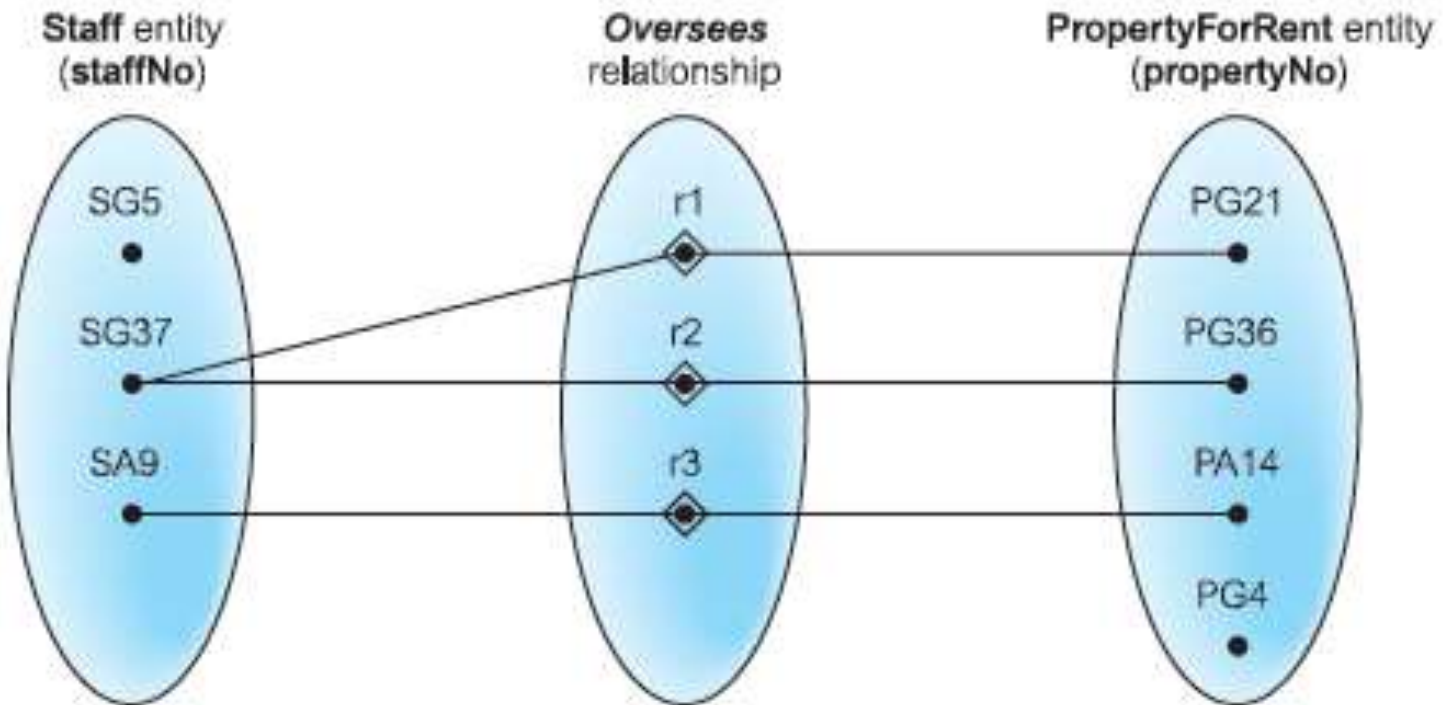


The multiplicity of the Staff *Manages* Branch one-to-one (1:1) relationship

# One-to-Many (1:\*) Relationships

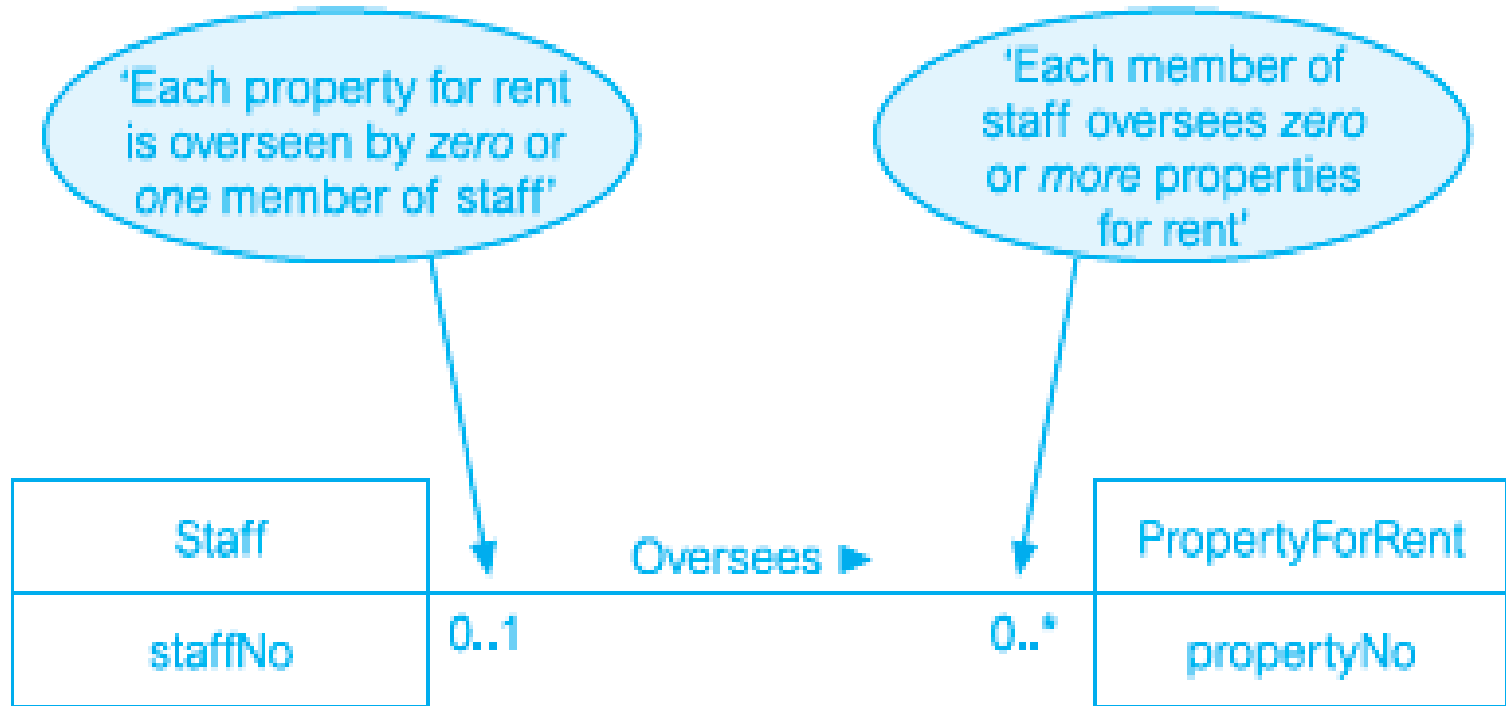
- For an entity type occurrence there are many entity type occurrences
- A member of staff oversees properties for rent (1:\*)

# One-to-Many (1:\*) Relationships



Semantic net showing three occurrences of the Staff *Oversees* PropertyForRent relationship type.

# Diagrammatic representation of 1:\* relationships

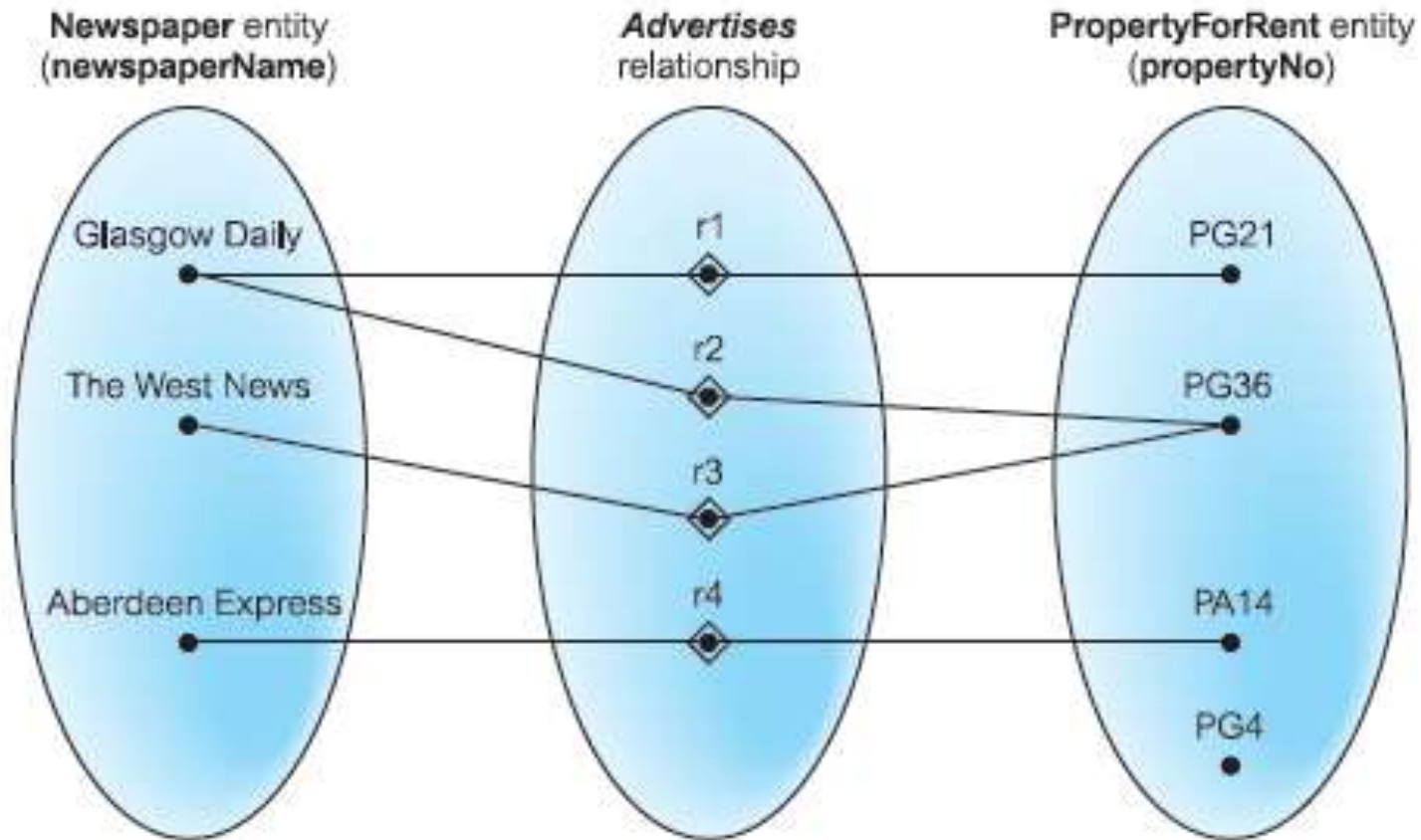


The multiplicity of the Staff *Oversees* PropertyForRent one-to-many (1:\*) relationship type

# Many-to-Many (\*:\*) Relationships

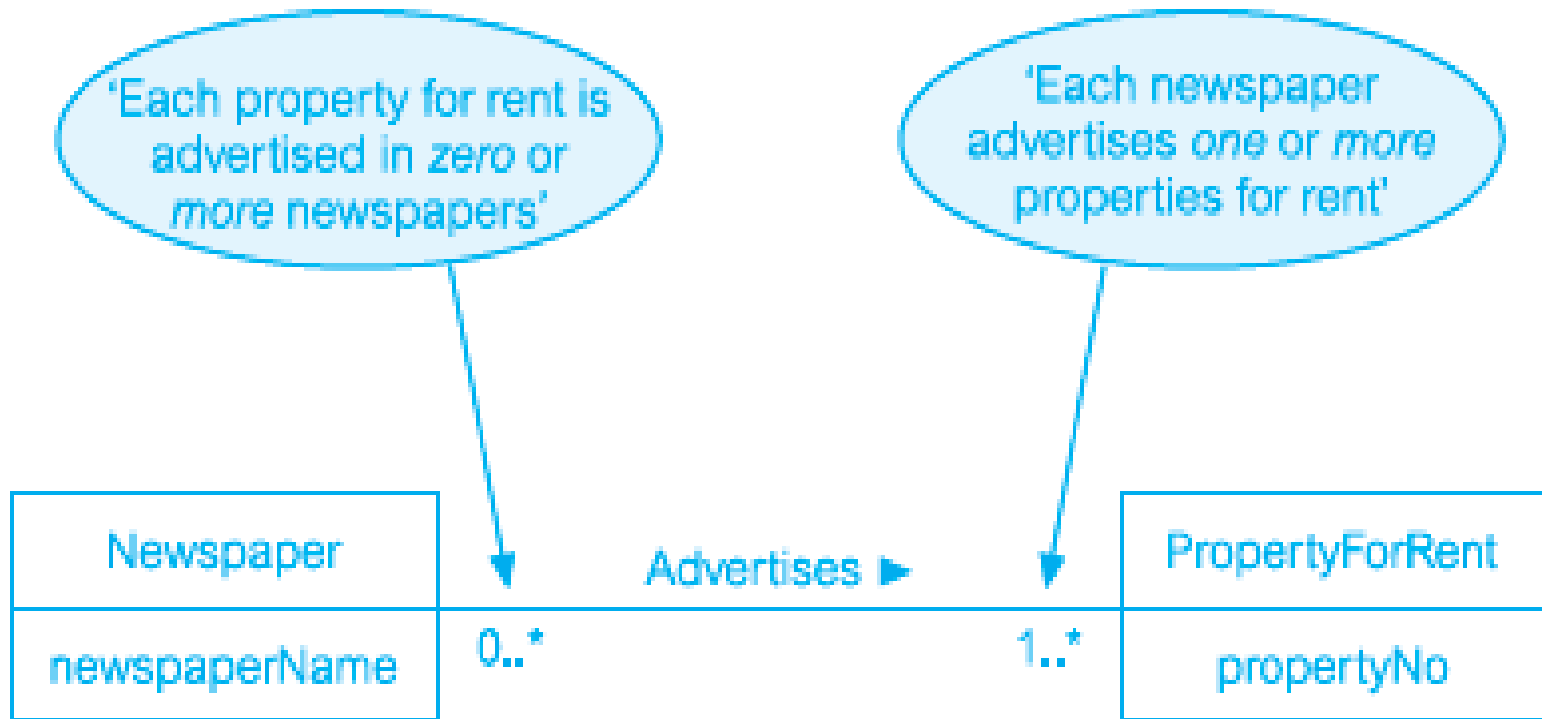
- For an entity type occurrence there are many entity type occurrences on both sides of the relationship
- Newspapers advertise properties for rent (\*:\*)

# Many-to-Many (\*:\*) Relationships



Semantic net showing four occurrences of the Newspaper *Advertises* PropertyForRent relationship type

# Diagrammatic representation of \*:\* relationships



The multiplicity of the Newspaper *Advertises* PropertyForRent many-to-many (\*:\*) 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

---



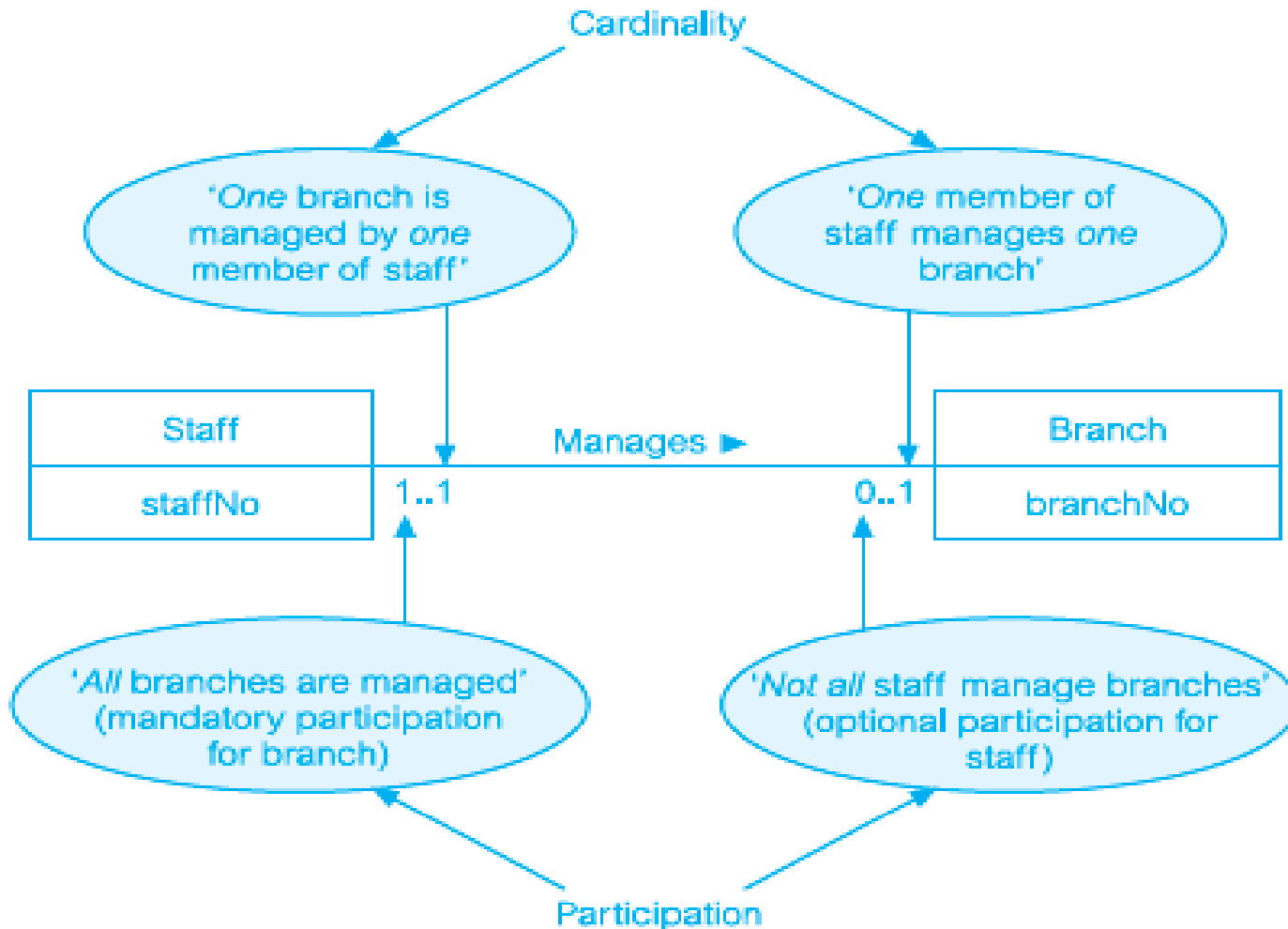
# Multiplicity

- **Made up of two types of restrictions on relationships:**
  - cardinality
  - and participation

# Multiplicity

- **Cardinality**
  - Describes the *maximum* number of possible relationship occurrences for an entity participating in a given relationship type
- **Participation**
  - Describes the *minimum* number of possible relationship occurrences for an entity participating in a given relationship type

# Multiplicity as cardinality and participation constraints



# Participation (Mandatory or Optional)

- The participation constraint represents the presence of at least one entity occurrence involved in a particular relationship (referred to as **mandatory** participation) or absence of at least one (referred to as **optional** participation)
- Optional participation is represented as a minimum value of 0 while mandatory participation is shown as a minimum value of 1