

Entity-Relationship Modeling

Objectives

At the end of this lesson, you should be able to:

- Define 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

Objectives

- 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

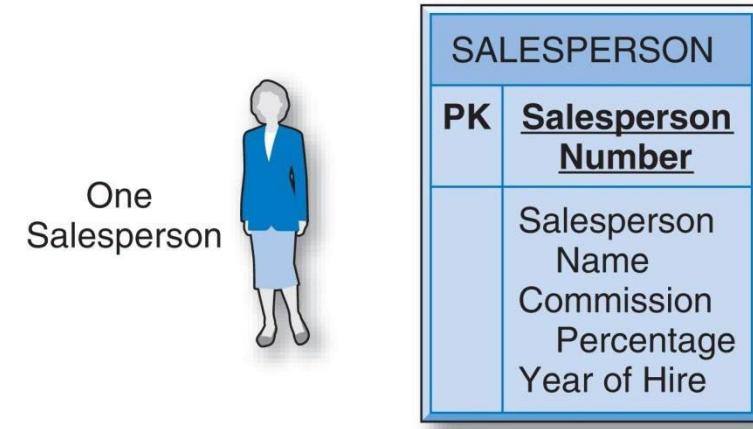
- ERM is existed as to explore the different ways that entities can relate to each other as they always do in the real world.
- ERM also is used to devise a way of recording, diagramming the entities and the ways in which they interrelate in the business environment.
- The things that you need to know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

Entity Types

- **Entity type** - A group of objects with the same properties, which are identified by the enterprise as having an independent existence.
 - ✓ Physical (real) existence [Person, Place, Thing]
 - ✓ Conceptual (abstract) existence [Event, Concept]

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

- **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

- A 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

Branch entity
(branchNo)

Has ►
relationship

Staff entity
(staffNo)

B003

B007

r1

r2

r3

SG37

SG14

SA9

Relationship
name

Staff

Branch

'Branch has staff'

Semantic Net View

Diagrammatic View

Relationship type: **HAS** → Branch HAS STAFF

Each occurrence of the HAS relationship associates **one** Branch entity occurrence
with **one** Staff entity occurrence

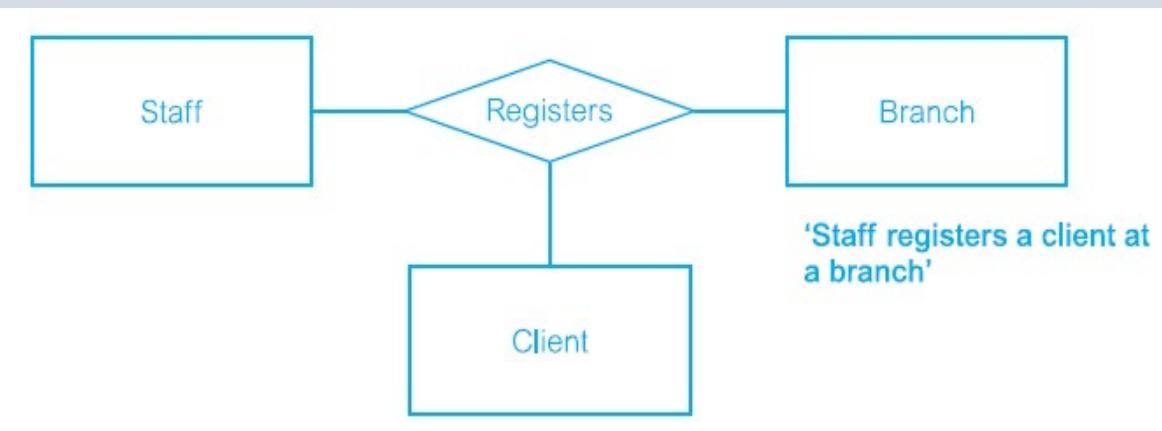
Degree of Relationship

- Refers to the number of participating entity types in a relationship.
- Relationship of degree :
 - ✓ Two is **binary**
 - ✓ Three is **ternary**
 - ✓ Four is **quaternary**

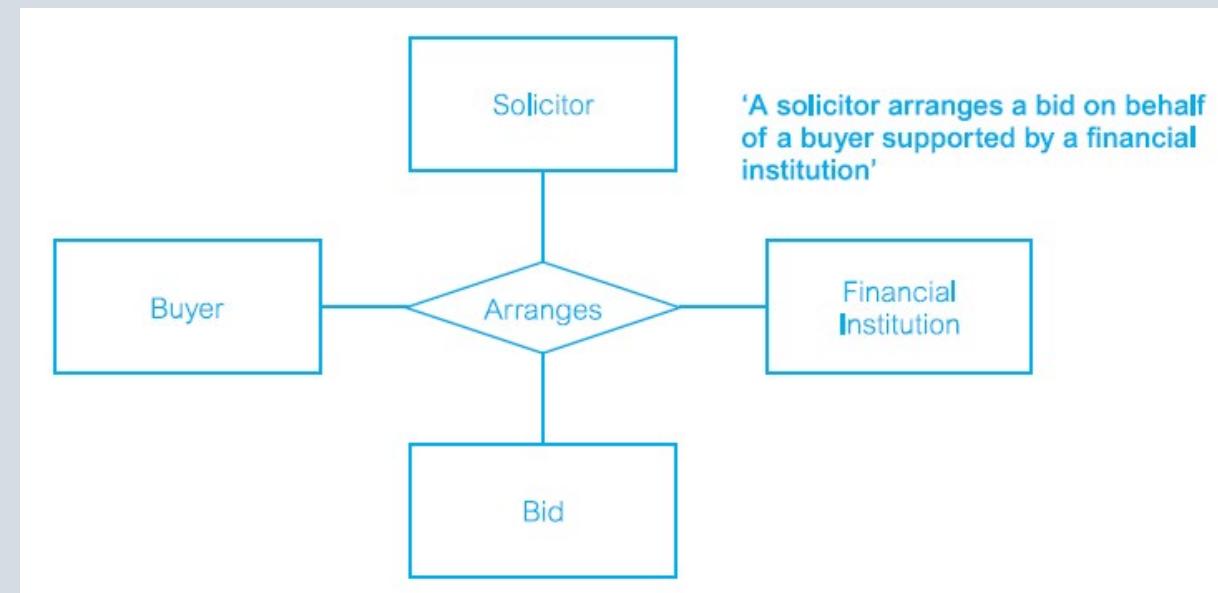
Binary Relationship



Ternary Relationship

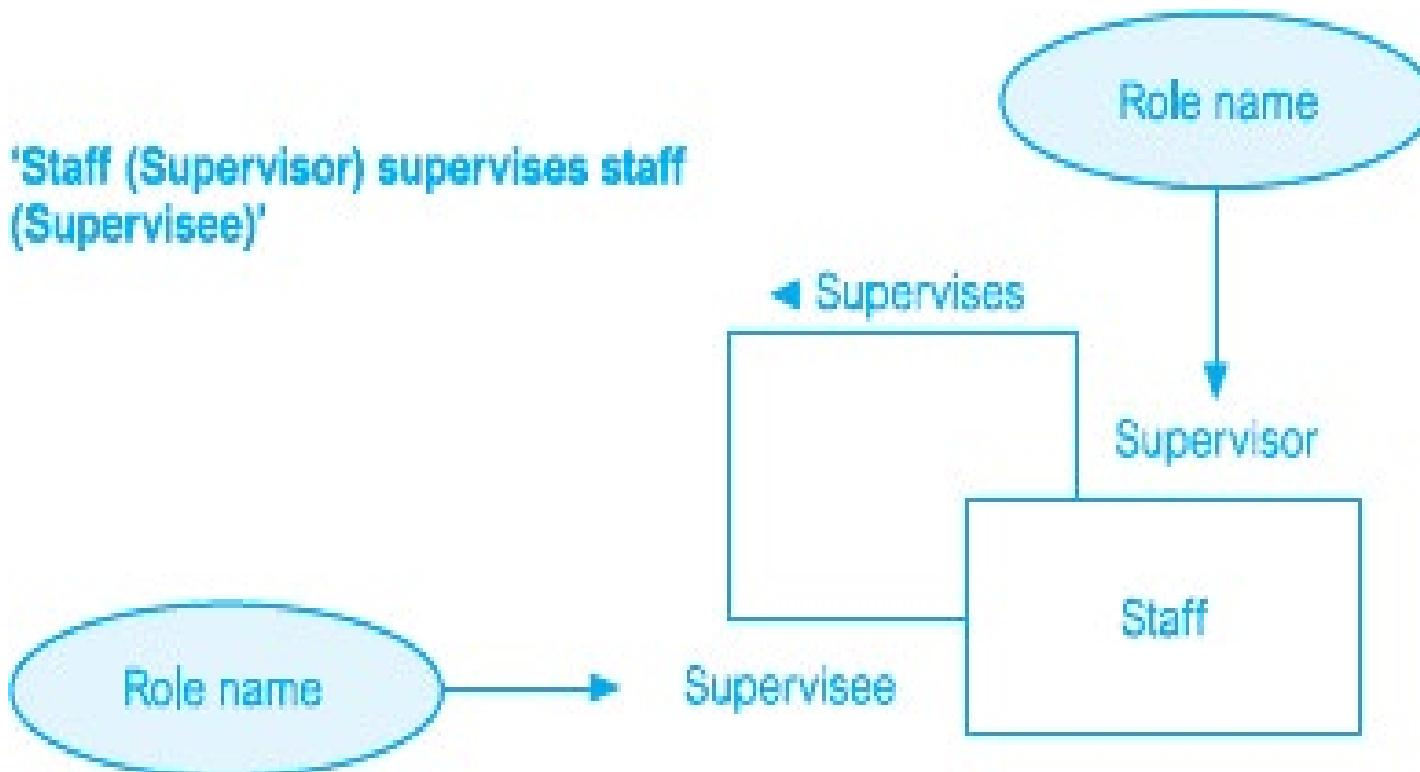


Quaternary Relationship

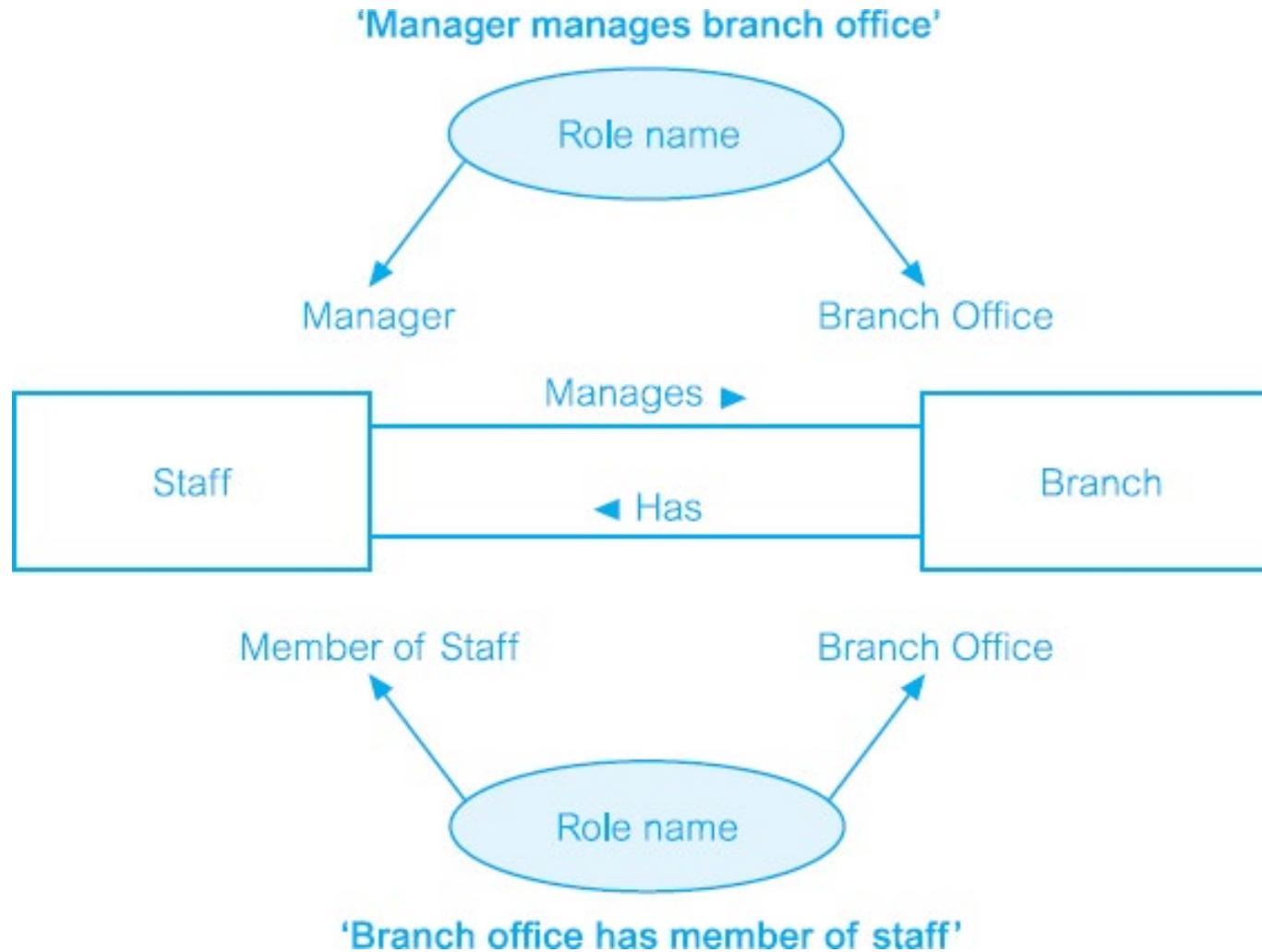


Recursive Relationship

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



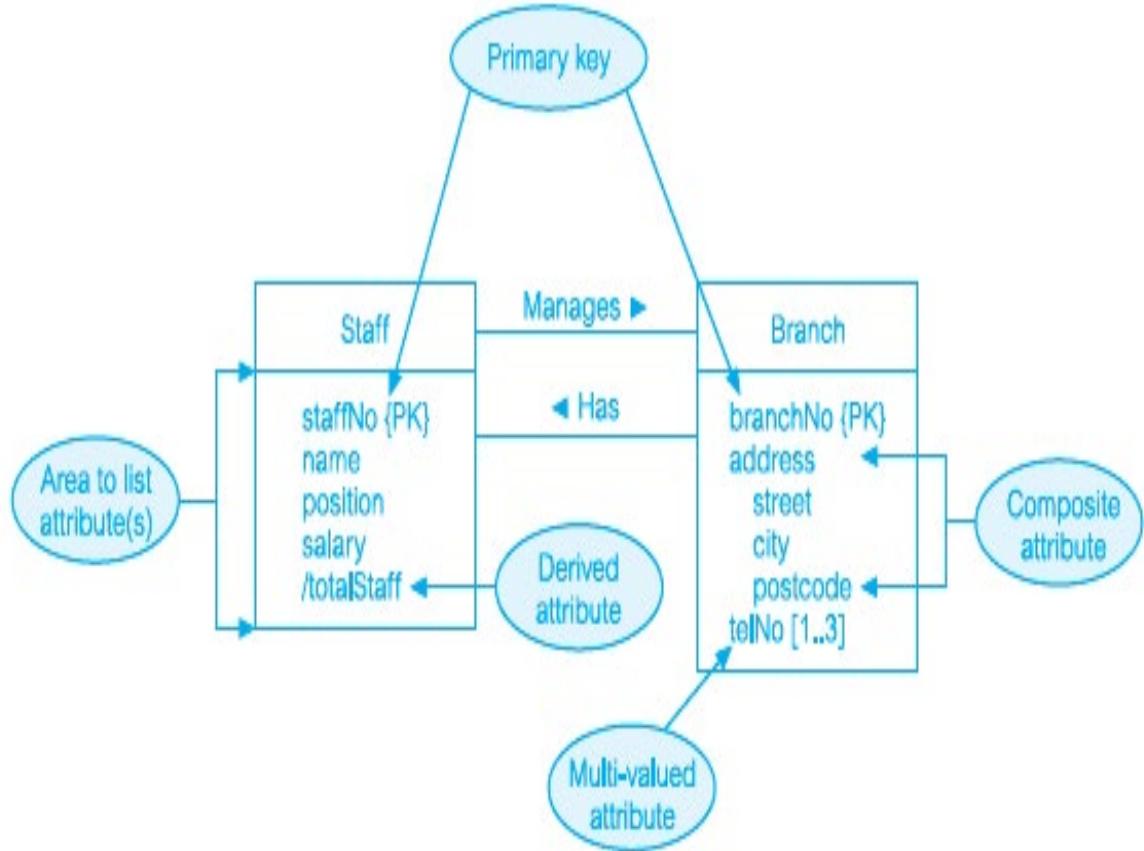
Entities associated through two distinct relationships with role names



Attributes

- **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

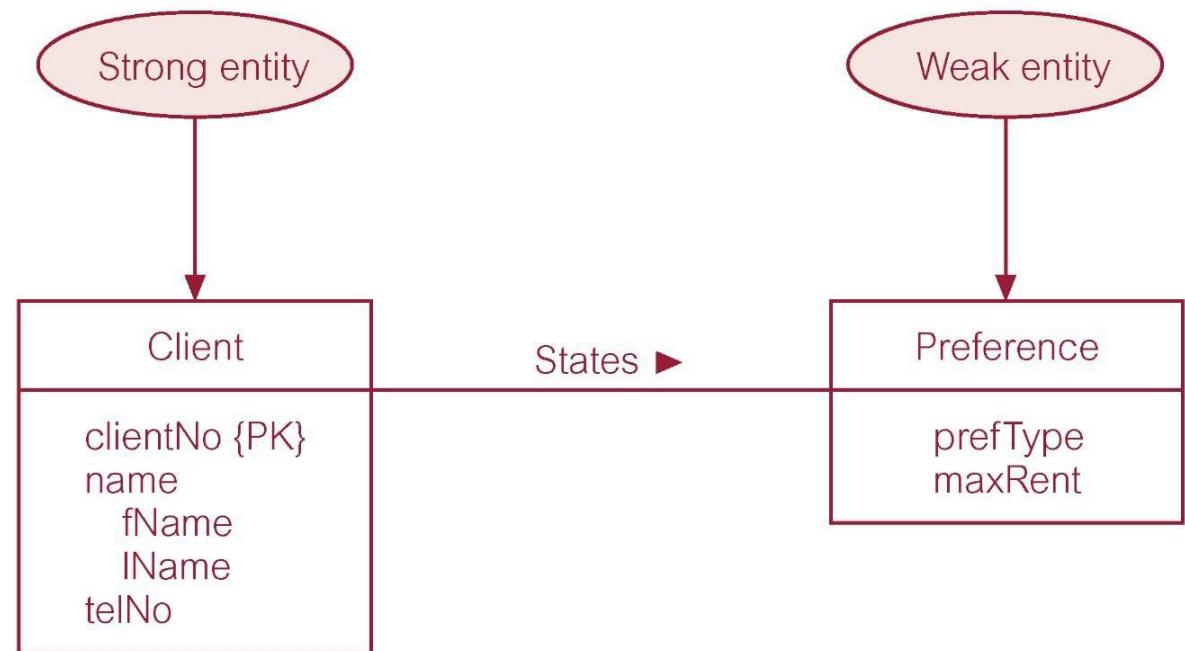
- **Simple attribute** - An attribute composed of a **single component** with an independent existence.
 - ✓ It cannot be further subdivided into smaller components. For example, Salary.
- **Composite attribute** - An attribute composed of **multiple components**, each with an independent existence.
 - ✓ 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 entity type:** An entity type that is existence-dependent on some other entity type.



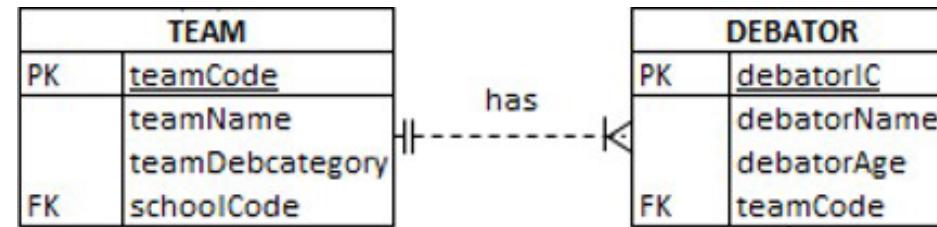
Strong and Weak Entity

Strong (Regular) entity

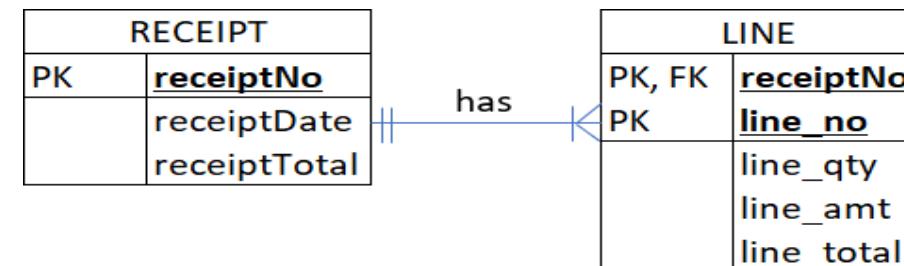
- Exists independently
- Has its own primary key

Weak entity

- Existence dependence
- Primary key is totally or partially derived from other entity

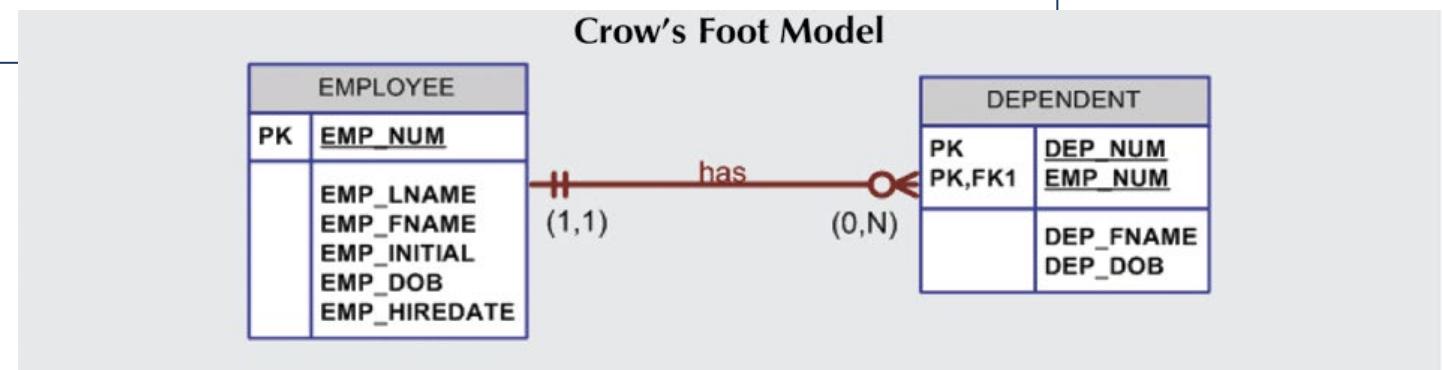


Weak entity



Weak Entity

- Weak entity meets two conditions
 1. **Existence-dependent**
 - Cannot exist without entity with which it has a relationship
 2. Has **primary key** that is **partially or totally derived** from parent entity in relationship
- Database designer usually determines whether an entity can be described as weak based on business rules



Weak Entity

Table name: EMPLOYEE

Database name: Ch04_ShortCo

		EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_DOB	EMP_HIREDATE
►	+	1001	Callifante	Jeanine	J	12-Mar-64	25-May-97
	+	1002	Smithson	William	K	23-Nov-70	28-May-97
	+	1003	Washington	Herman	H	15-Aug-68	28-May-97
	+	1004	Chen	Lydia	B	23-Mar-74	15-Oct-98
	+	1005	Johnson	Melanie		28-Sep-66	20-Dec-98
	+	1006	Ortega	Jorge	G	12-Jul-79	05-Jan-02
	+	1007	O'Donnell	Peter	D	10-Jun-71	23-Jun-02
	+	1008	Brzenski	Barbara	A	12-Feb-70	01-Nov-03

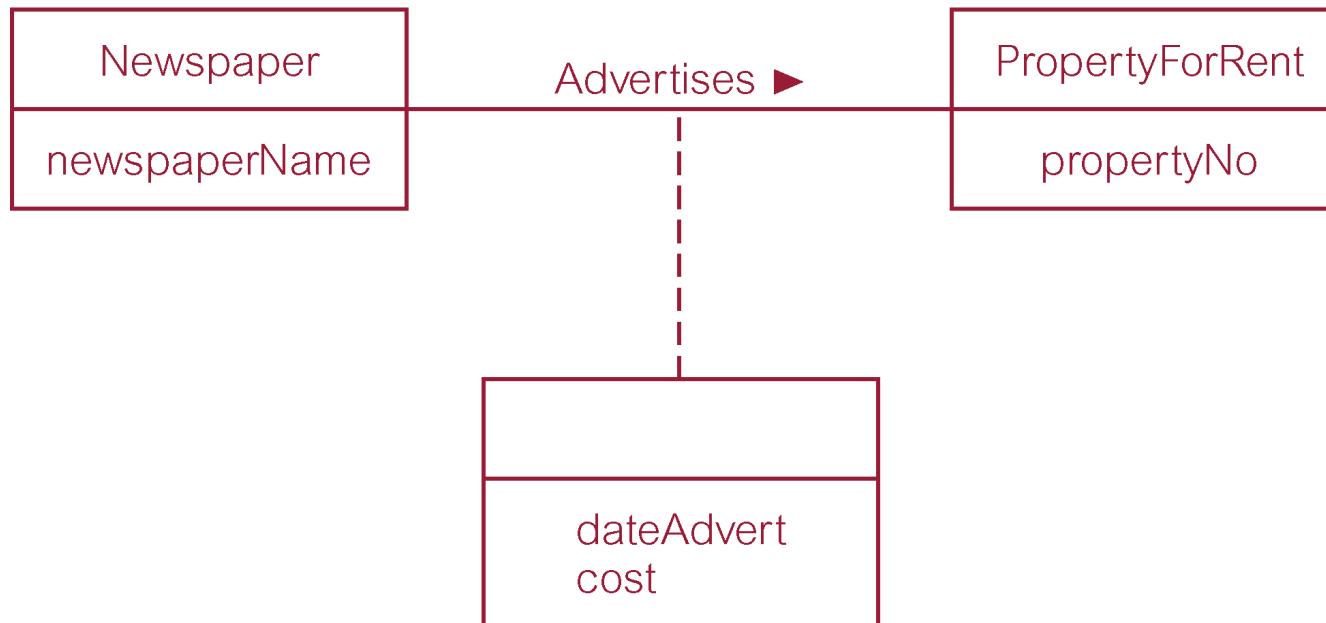
Table name: DEPENDENT

	EMP_NUM	DEP_NUM	DEP_FNAME	DEP_DOB
►	1001	1	Annelise	05-Dec-97
	1001	2	Jorge	30-Sep-02
	1003	1	Suzanne	25-Jan-04
	1006	1	Carlos	25-May-01
	1008	1	Michael	19-Feb-95
	1008	2	George	27-Jun-98
	1008	3	Katherine	18-Aug-03

Weak entity

Attributes on Relationship

‘Newspaper advertises property for rent’



Later will be converted as **bridge entity**

Structural Constraints

- Main type of constraint on relationships is called **multiplicity**.
 - ✓ Example of constraint: requirement that a property for rent must have an owner and each branch must have a staff
- **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:

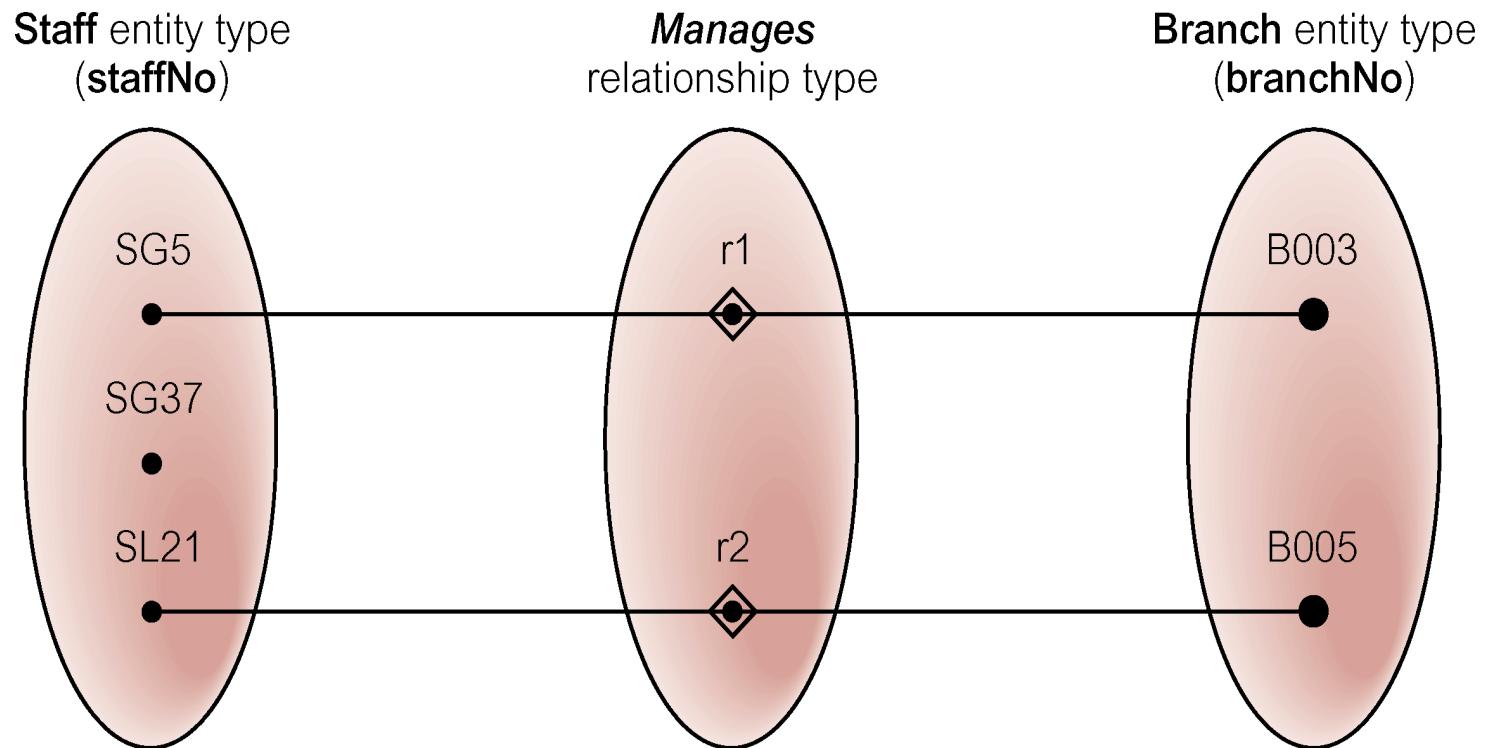
1. one-to-one (1:1)
2. one-to-many (1:*)
3. many-to-many (*:*)

A member of staff manages a branch

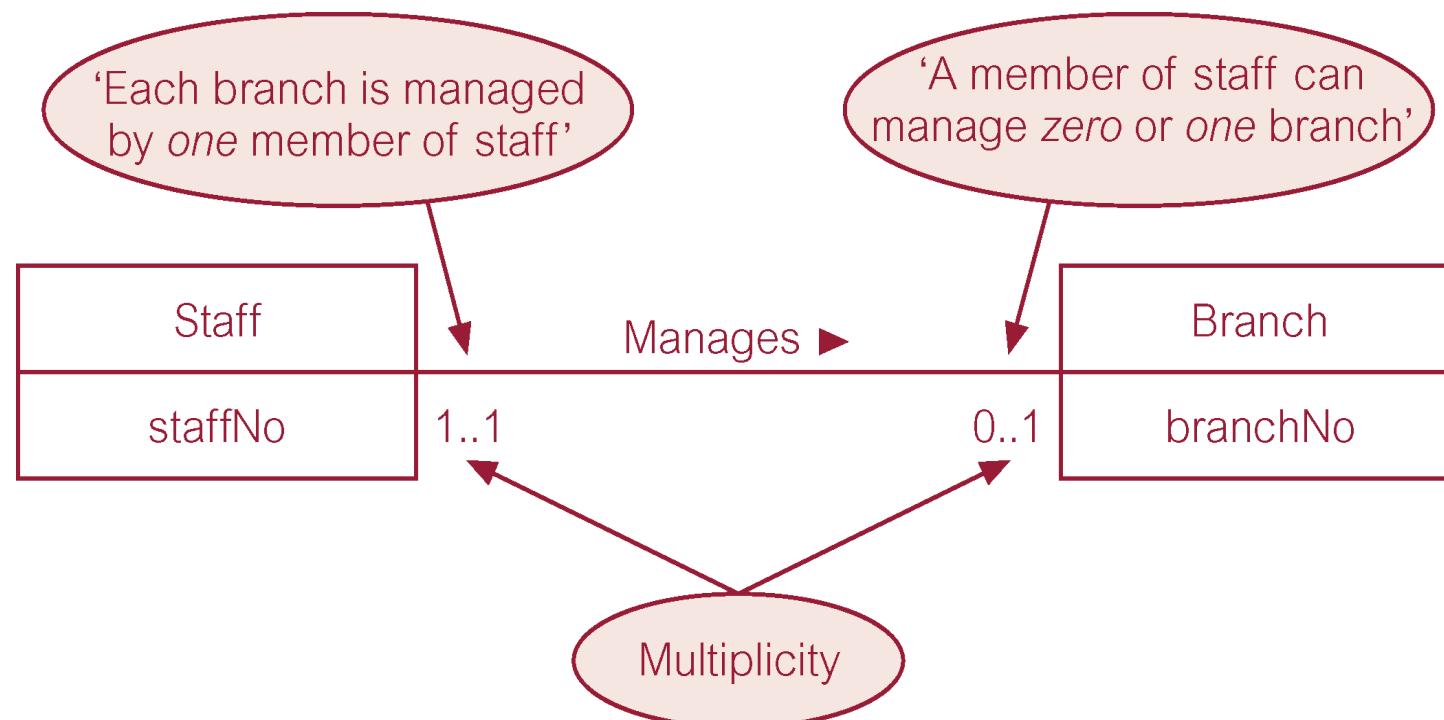
A member of staff oversees properties for rent

Newspapers advertise properties for rent

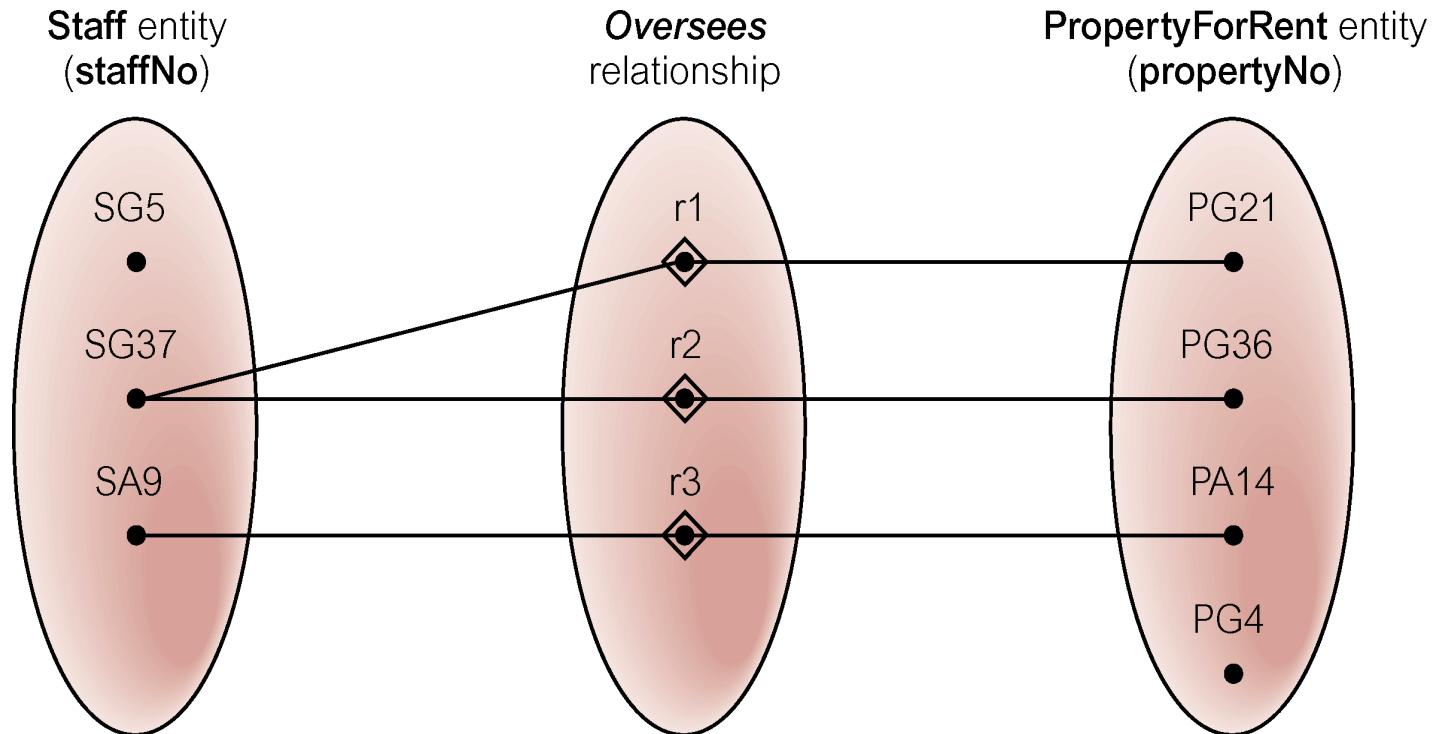
Semantic net of Staff *Manages* Branch relationship type



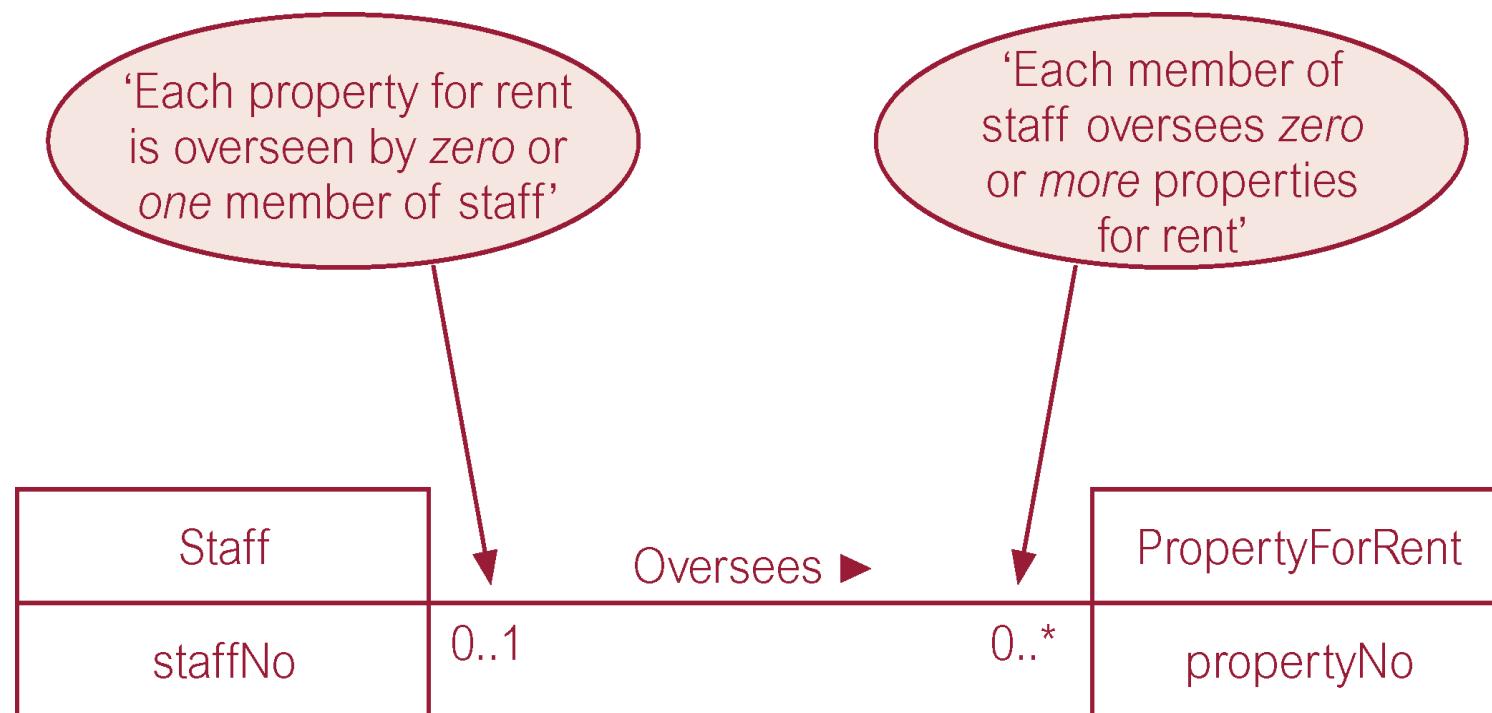
Multiplicity of Staff Manages Branch (1:1)relationship



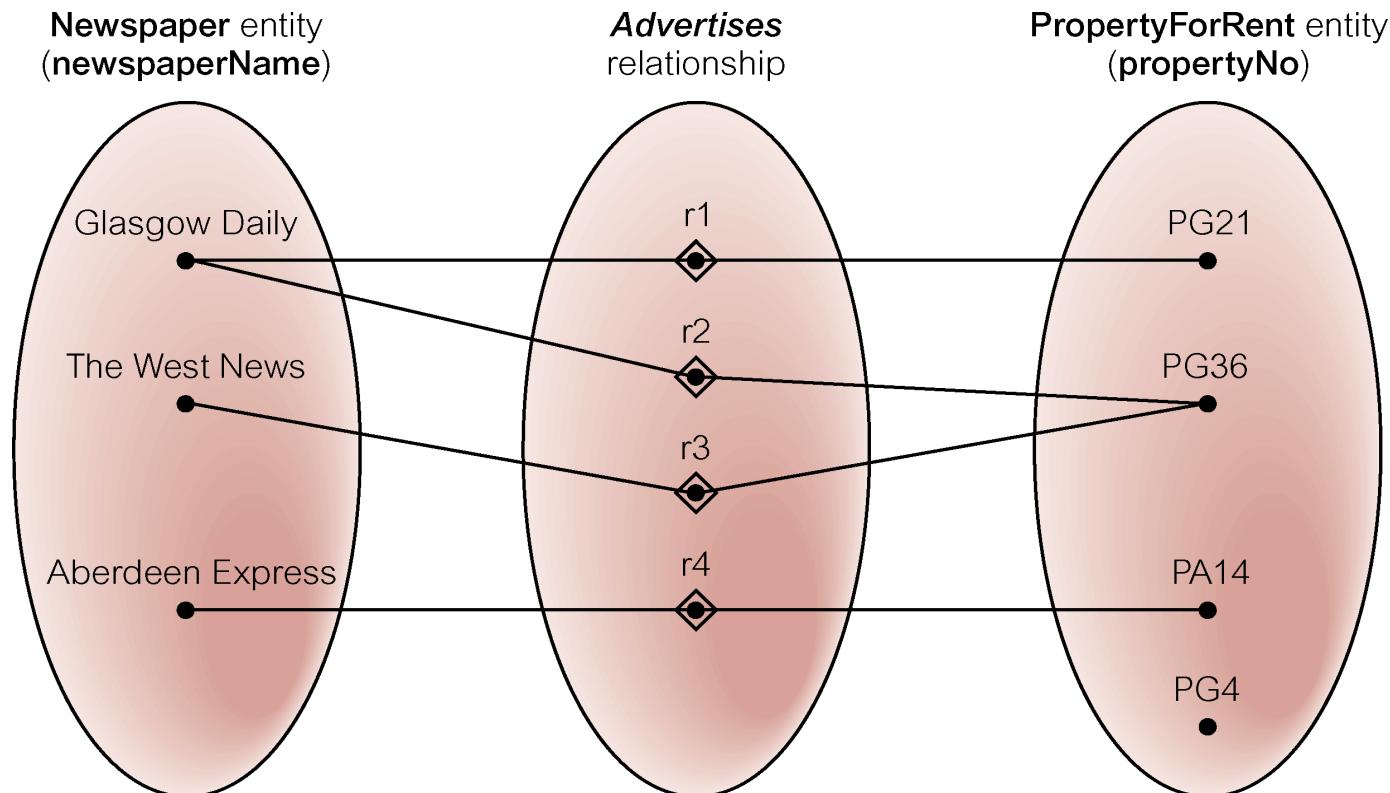
Semantic net of Staff Oversees PropertyForRent relationship type



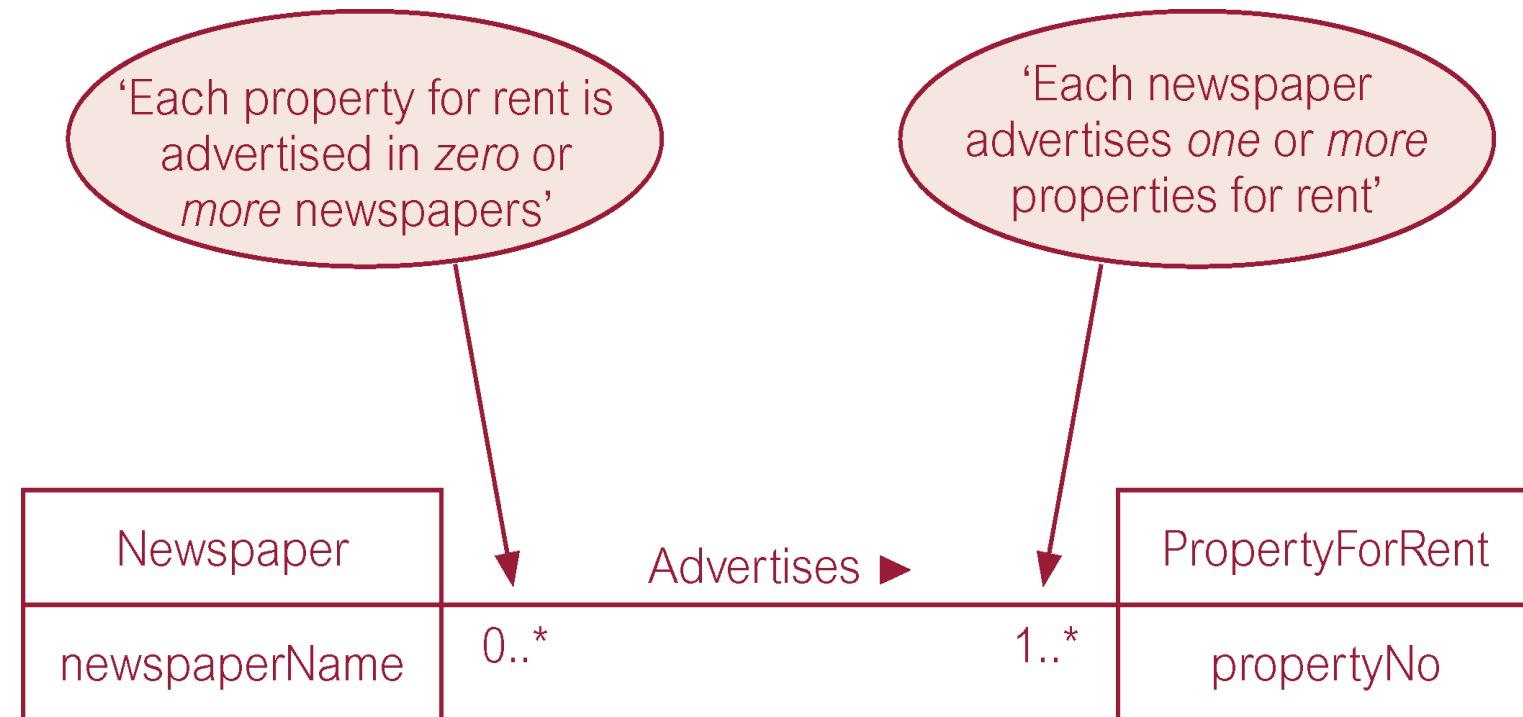
Multiplicity of Staff *Oversees* PropertyForRent (1:*) relationship type



Semantic net of Newspaper Advertises PropertyForRent relationship type



Multiplicity of Newspaper Advertises PropertyForRent (*:*) relationship



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**.

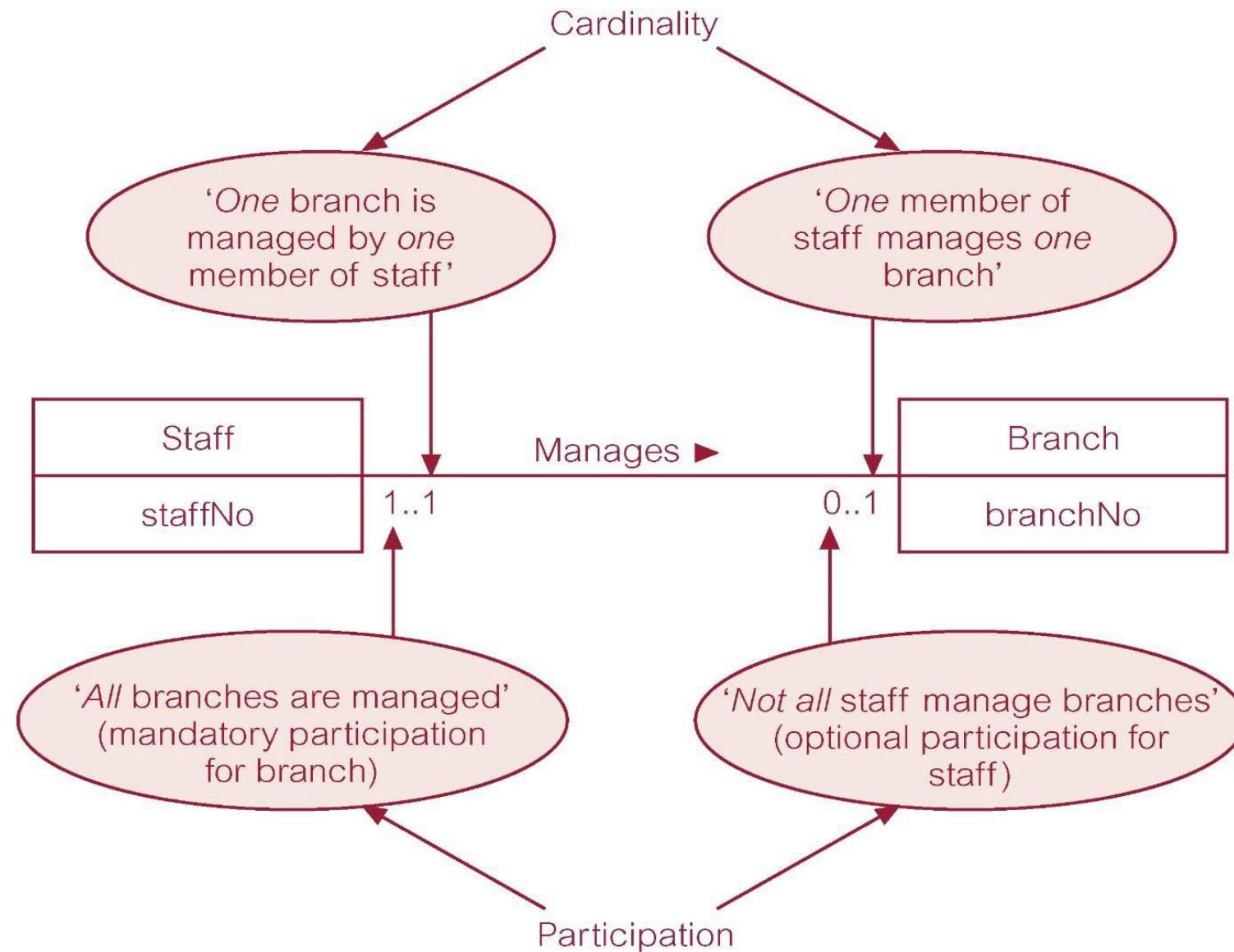
Cardinality

- 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

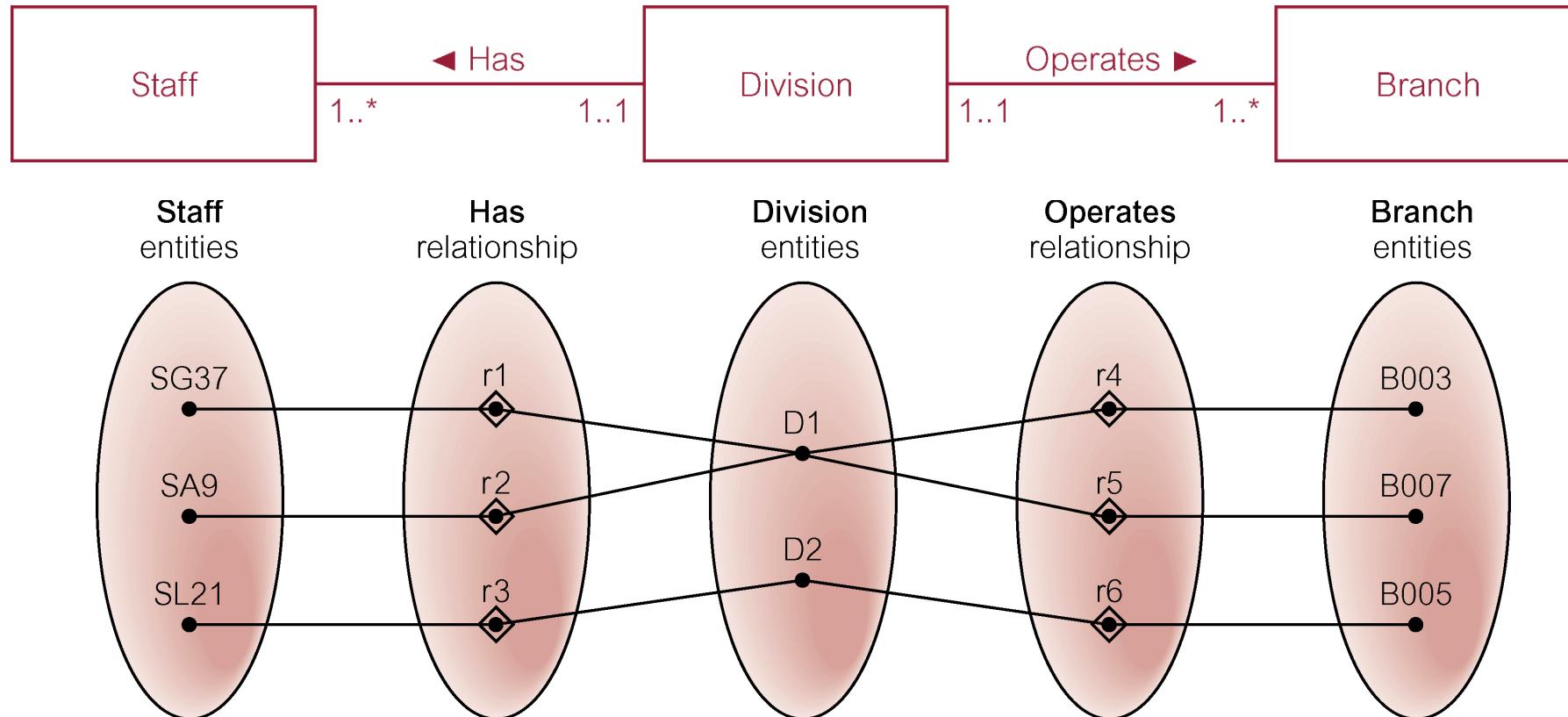


Problems with ER Modeling

- 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.

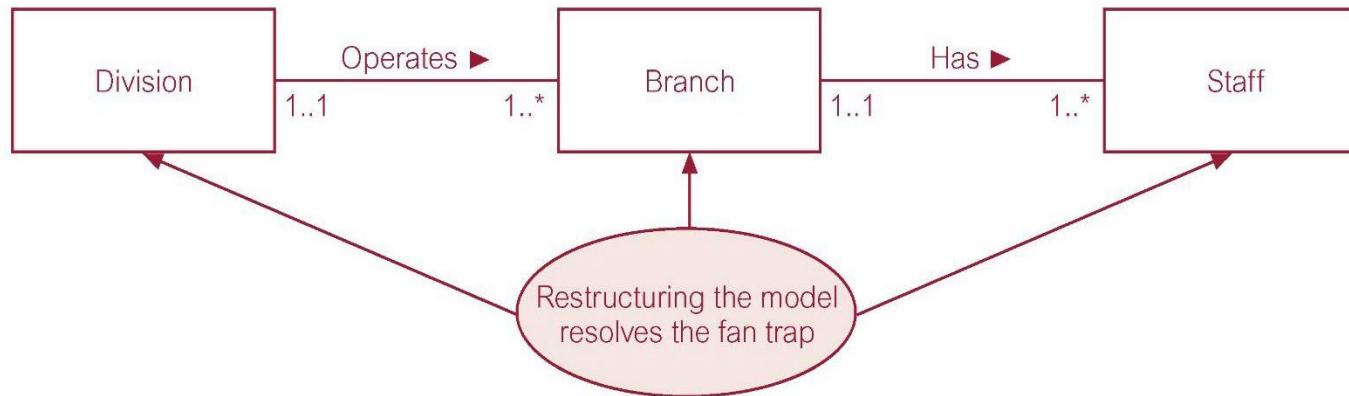
Fan Trap

A fan trap may exist where **two or more 1:*** relationships fan out from the same entity.

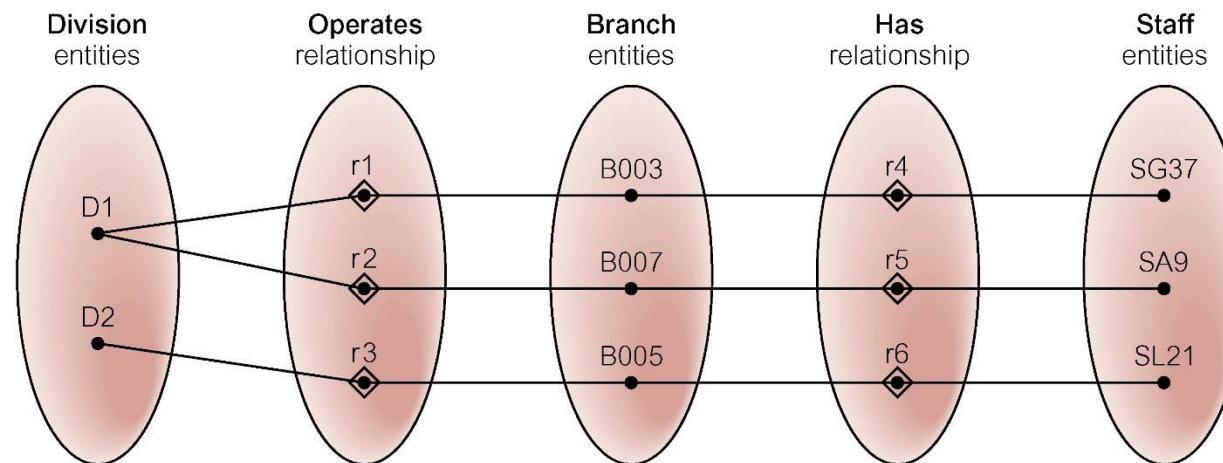


- At which branch office does staff number SG37 work?
SG37 works at Branch B003 or B007
- Inability to answer question, is the result of a fan trap associated with misrepresentation of relationship between Staff, Division & Branch entities.

Solution of Fan Trap



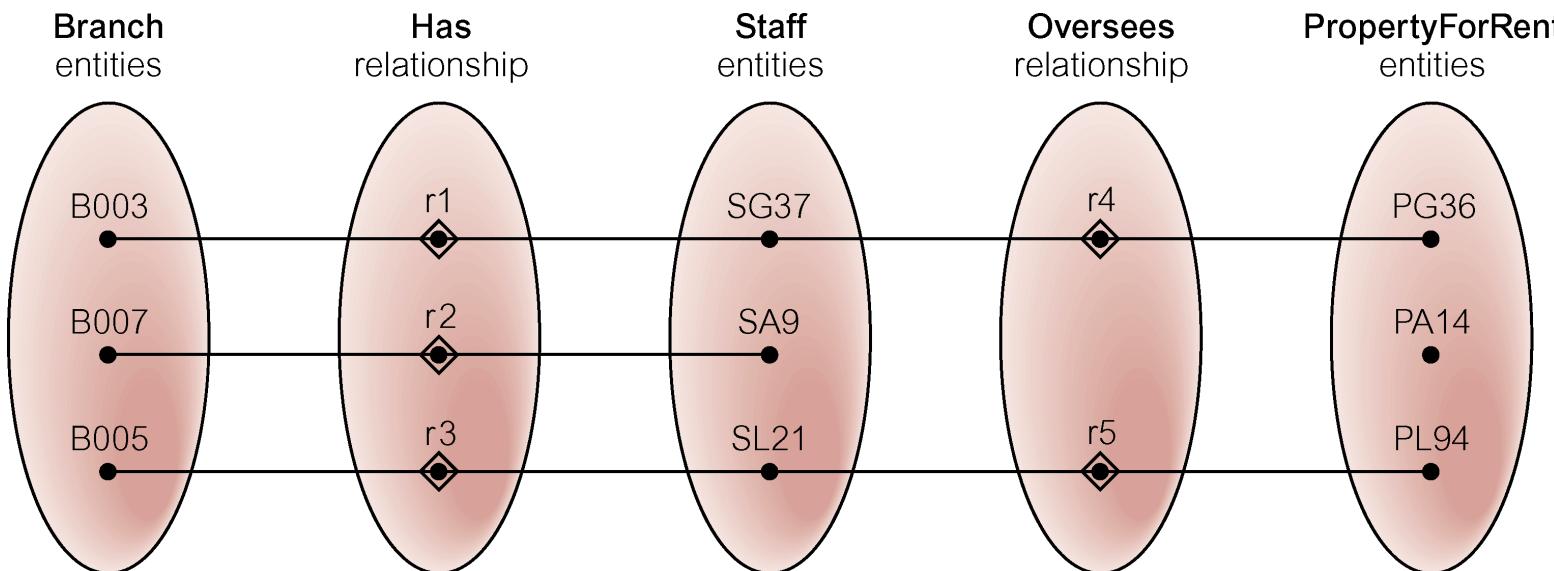
- Resolve fan trap by restructuring the original ER model to represent the correct association between these entities.



At which branch office does staff number SG37 work?
- SG37 works at branch B003

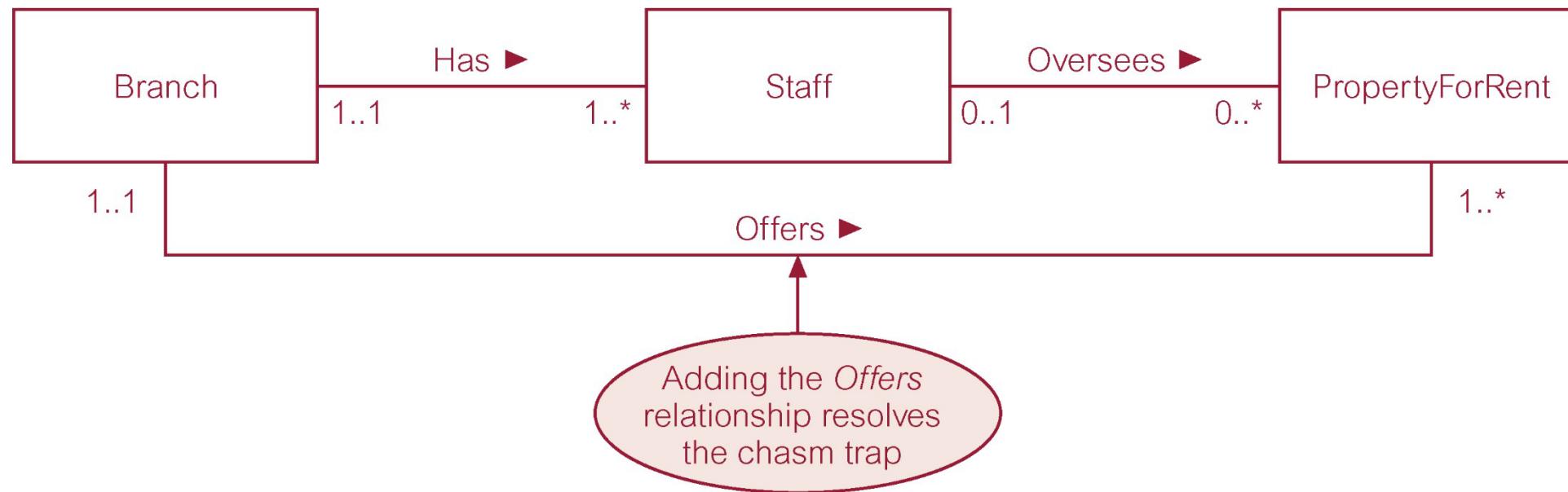
Chasm Trap

A chasm trap may occur where there are **one or more relationships with a minimum multiplicity of zero** (optional participation) forming part of the way between related entities.

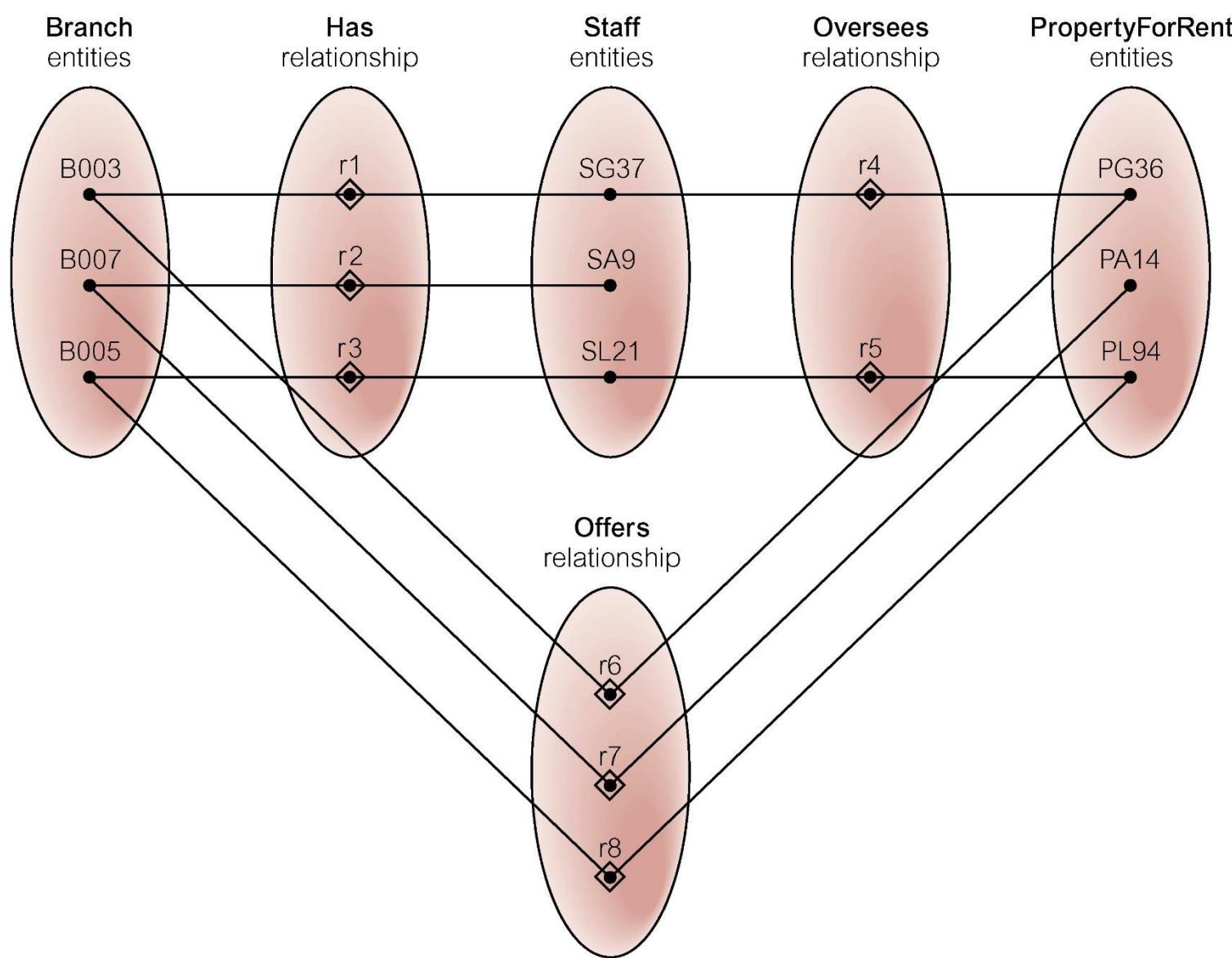


- At which branch office is property PA14 available?
- We unable to answer this question, as the properties is not yet allocated to a member of staff working at a branch.

Solution of Chasm Trap



- Resolve chasm trap by adding missing relationship (in this case) , *Offers* relationship between the Branch and PropertyForRent entities.



- At which branch office is property PA14 available?
- Answer: property PA14 is available at branch # B007

References

- Thomas Connolly and Carolyn Begg, Database Systems: A Practical Approach to Design, Implementation, and Management, 6th Edition, Pearson, 2015, ISBN: 978- 01329432