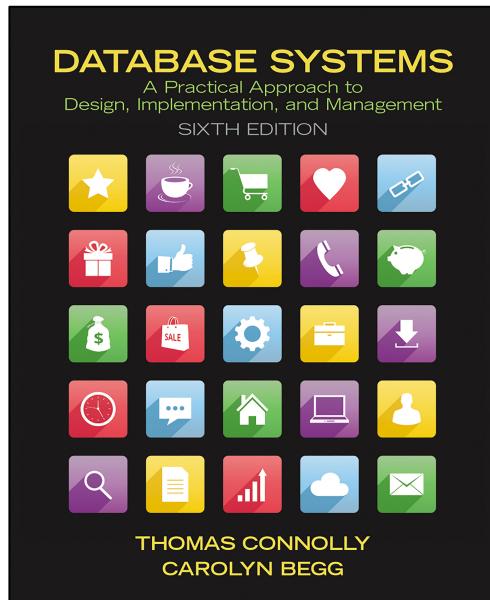

Structural constraints

Topic 1

Lesson 2 – Constraints on Relationships with UML notation

Chapter 12 Section 6 Connolly and Begg



Structural constraints

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 relationship.

Cardinality

Describes **maximum number** of possible relationship occurrences for an entity participating in each relationship type.

Participation

Determines whether **all or only some** entity occurrences participate in a relationship, describes **minimum number** of possible relationship occurrences

Represents policies (called **business rules**) established by user or company.

Cardinality constraint

The most common degree for relationships is binary.
Binary relationships are generally referred to as being:

one-to-one (1:1)

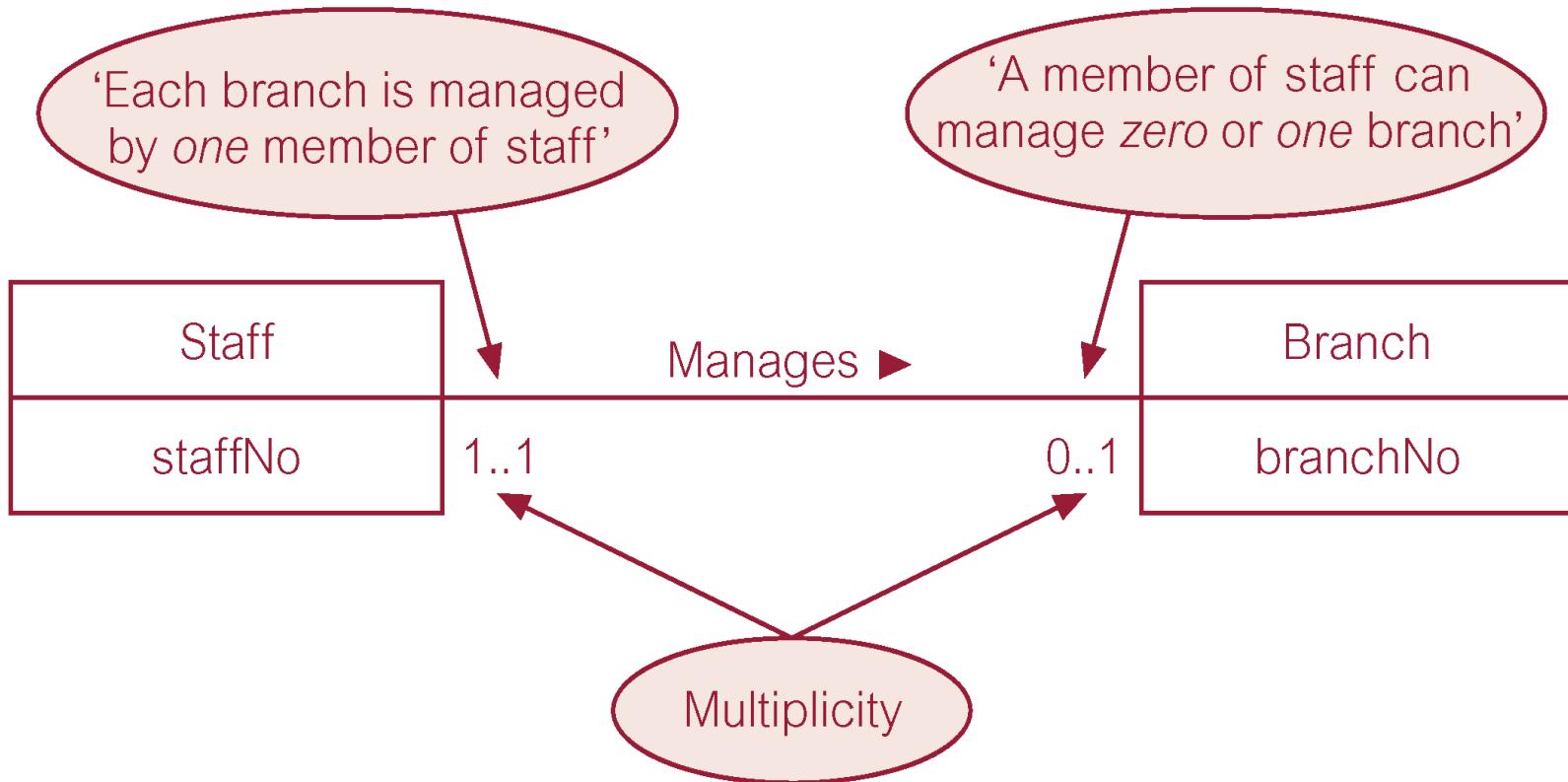
one to many (1:*)

many to many (*:*)

This representation is not representing the participation constraint

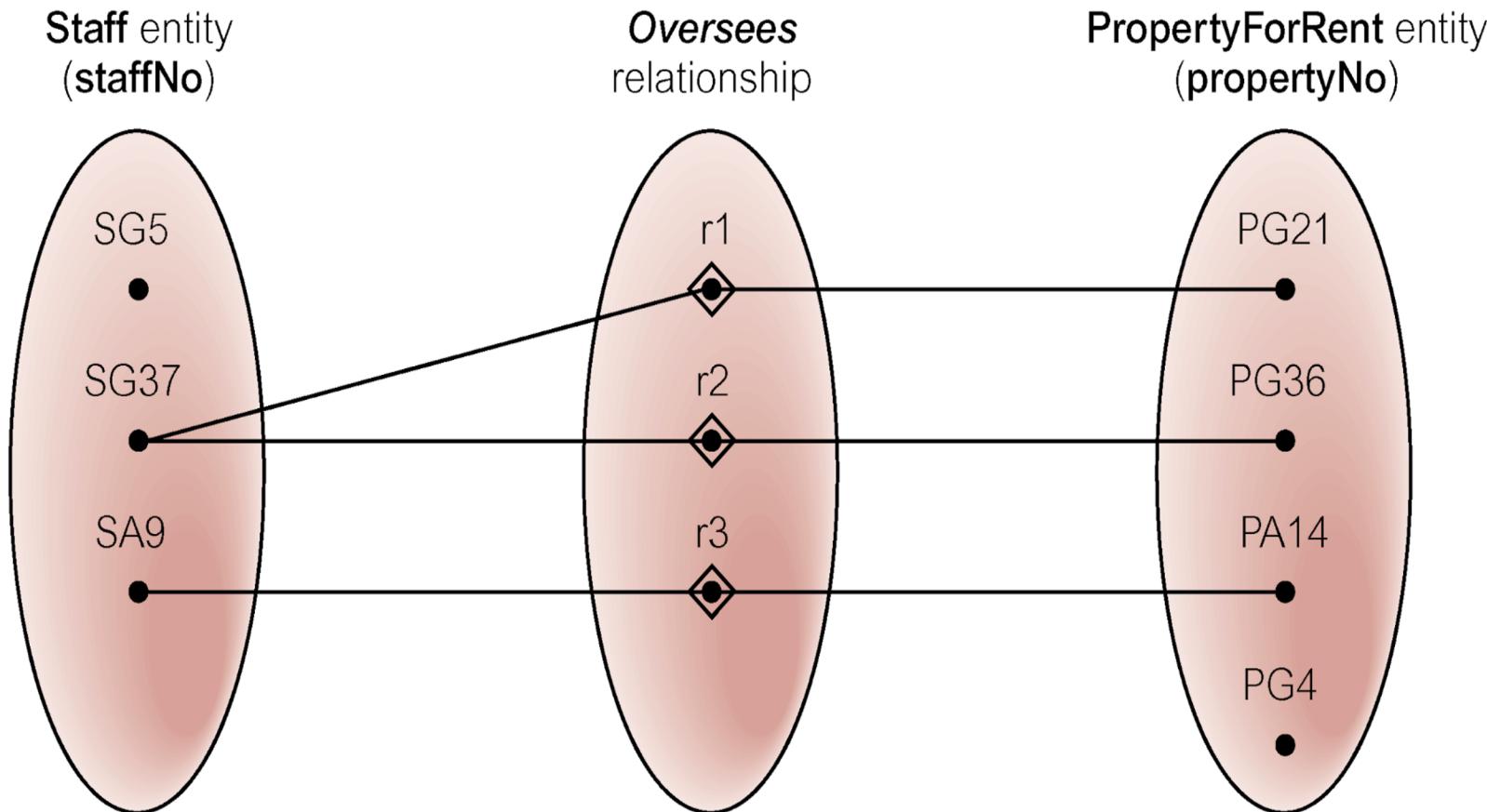
Multiplicity for Staff Manages Branch

Example of a 1:1 relationship



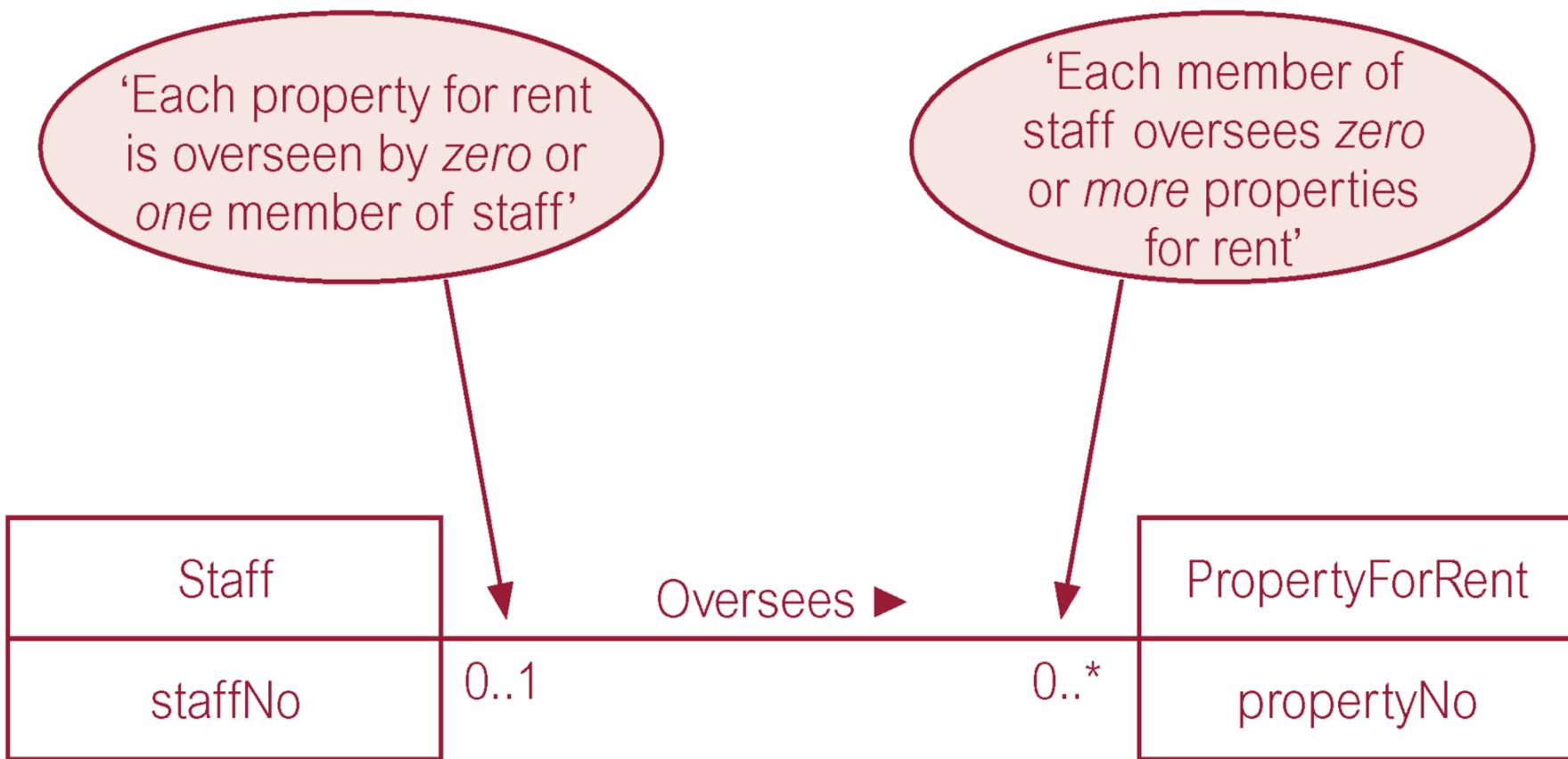
Semantic net of Staff *Oversees* PropertyForRent

What can we derive about the constraints?



Multiplicity of Staff *Oversees* PropertyForRent

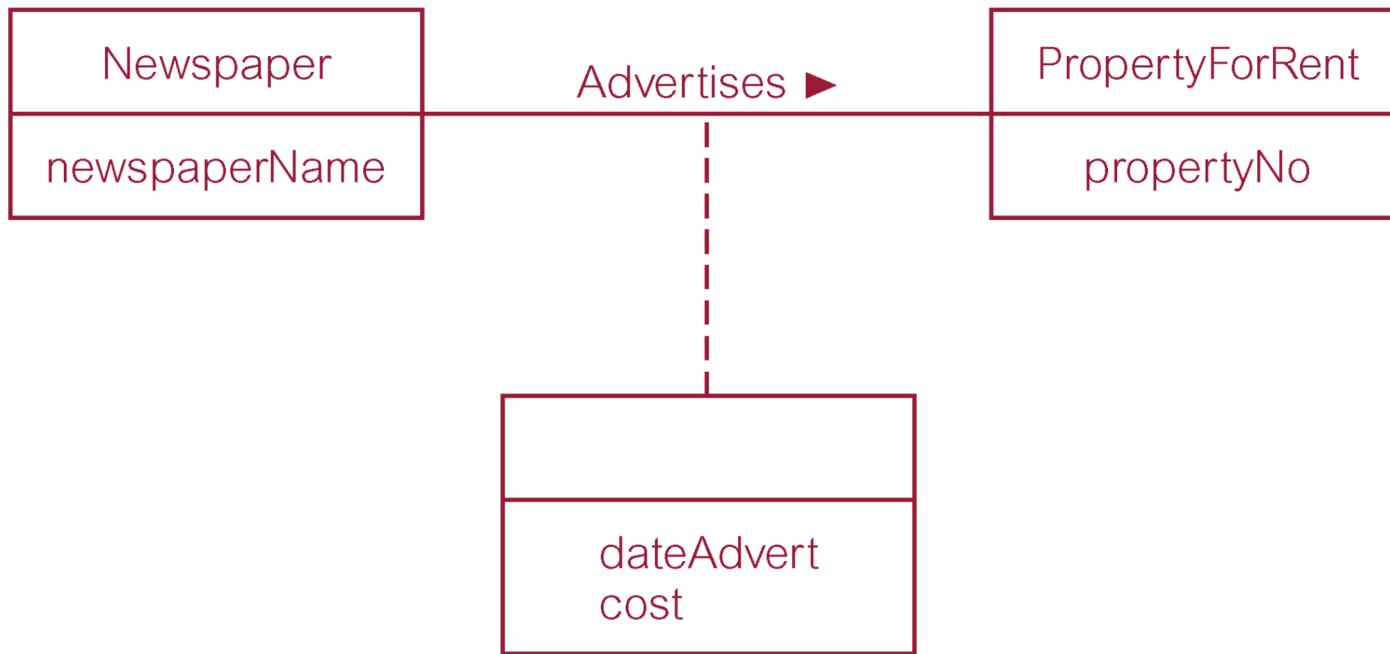
Specifying constraints on a (1:*) relationship type



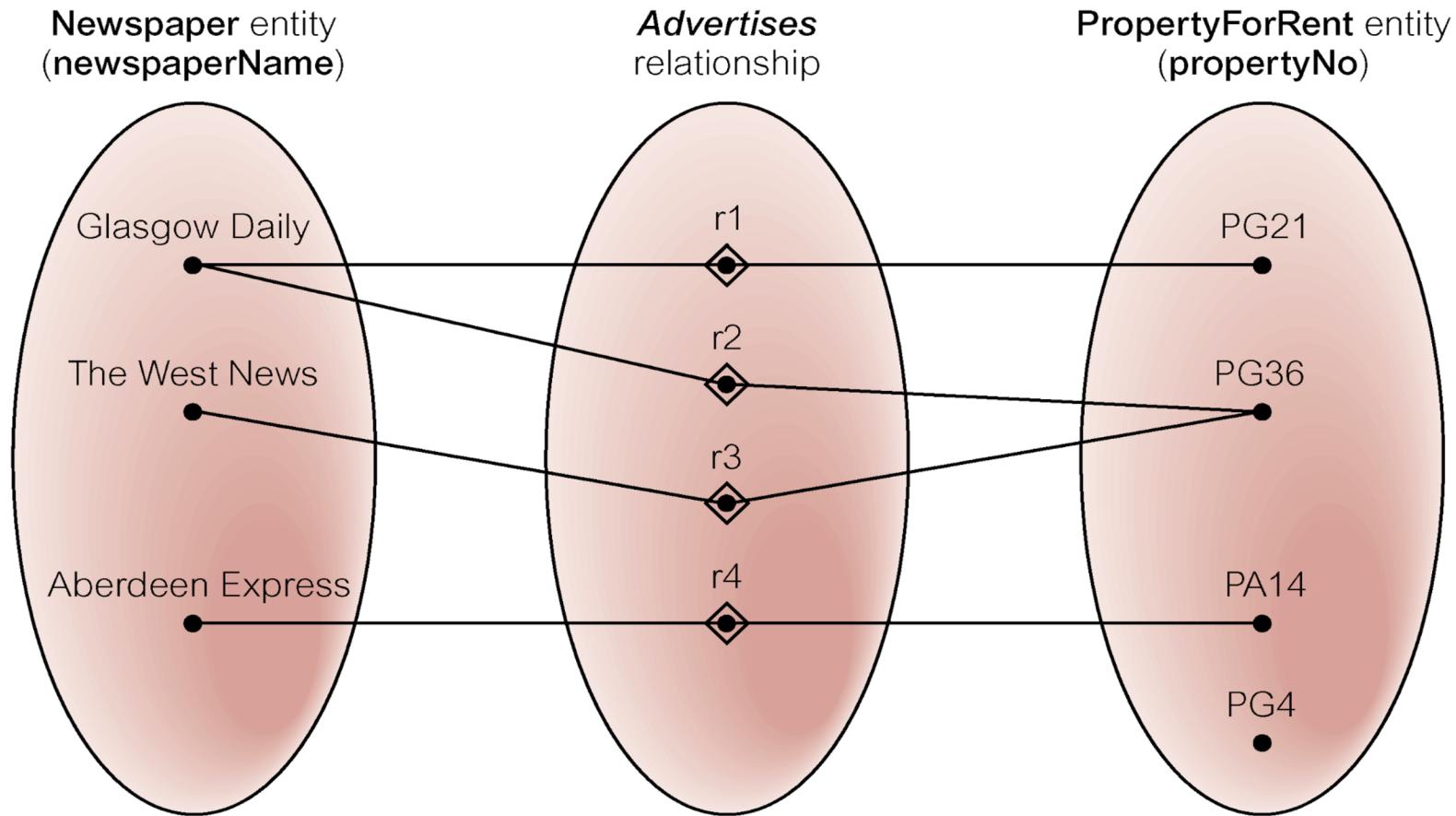
Exercise: identifying multiplicity

Identify the cardinality and participation constraints as well as the statements for each entity type in the Advertises relationship

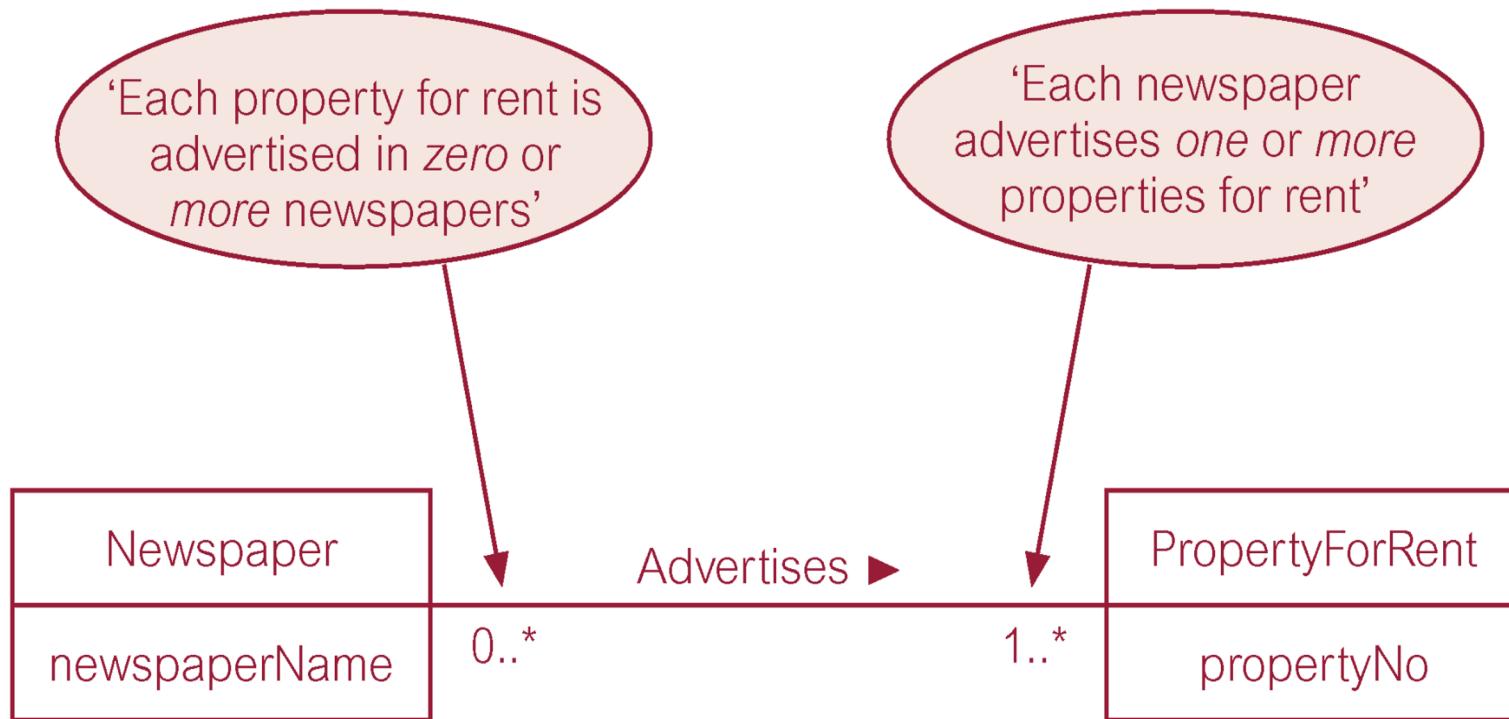
'Newspaper advertises property for rent'



Semantic net: Newspaper *Advertises* PropertyForRent

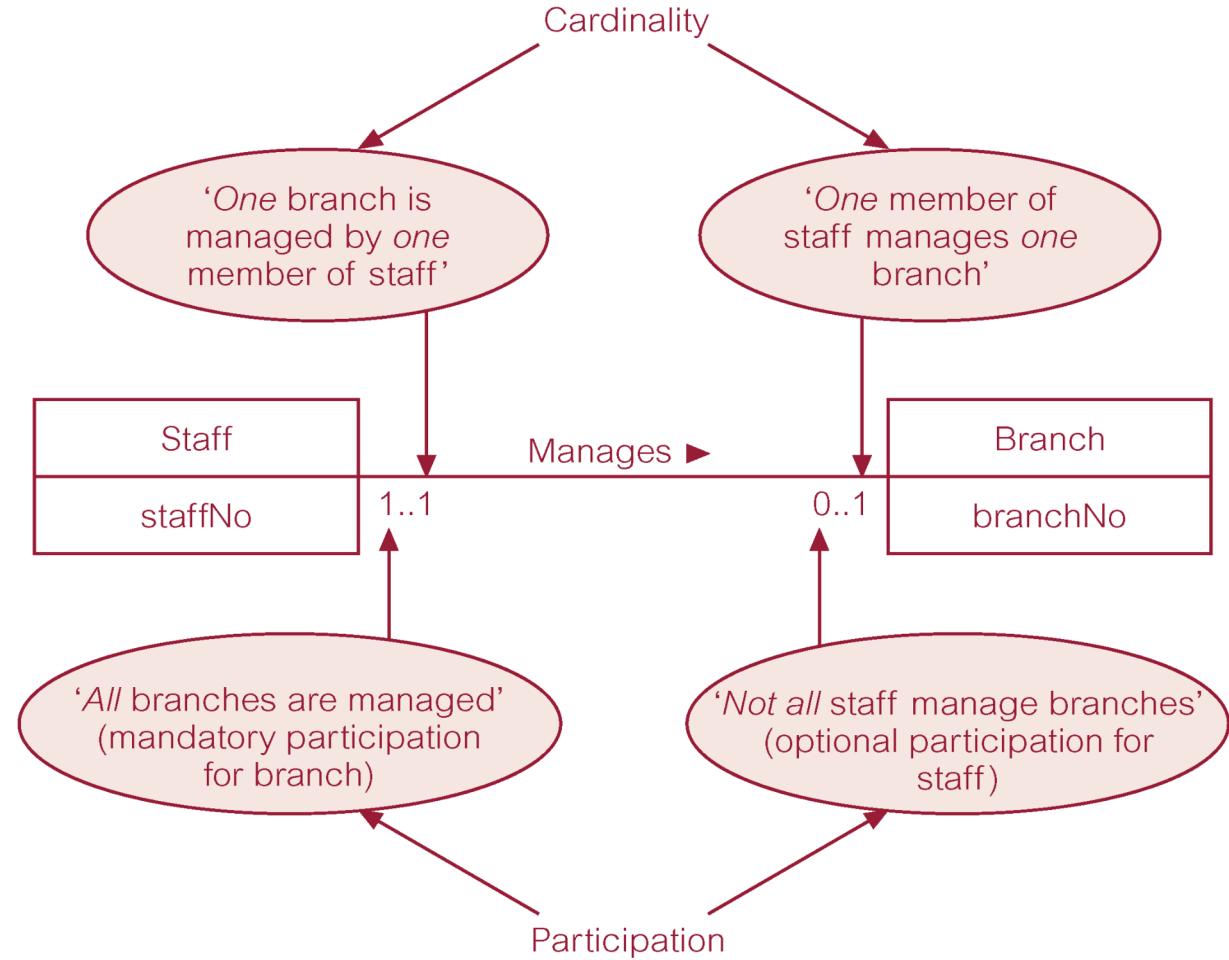


Solution: Multiplicity statements



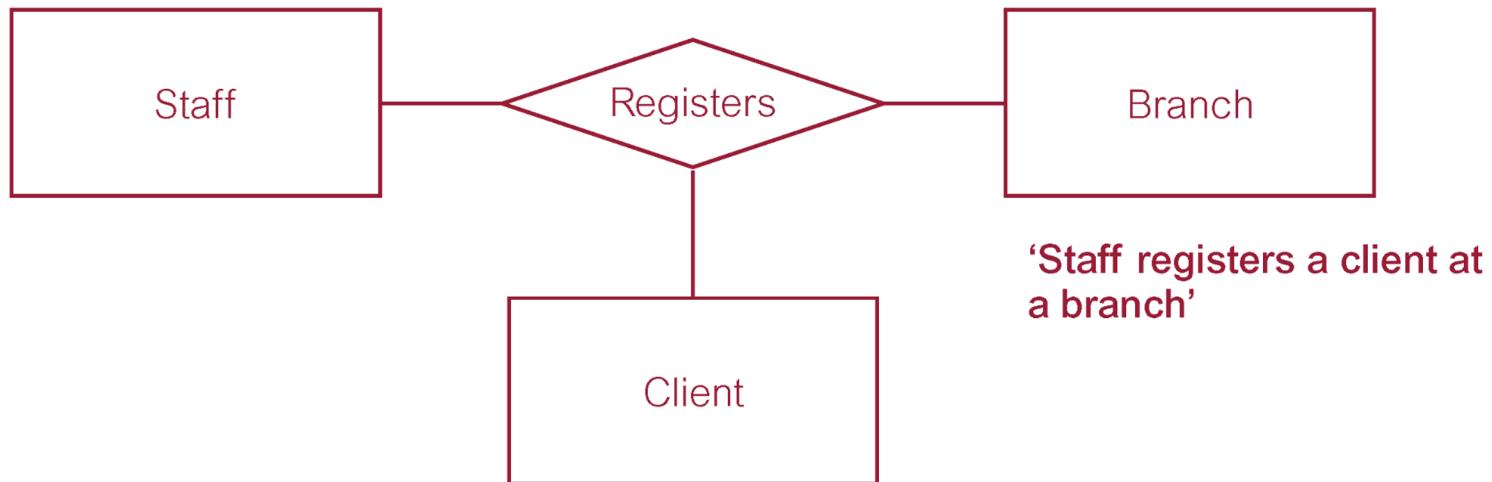
Multiplicity: participation and cardinality

Specifying constraints on the relationship



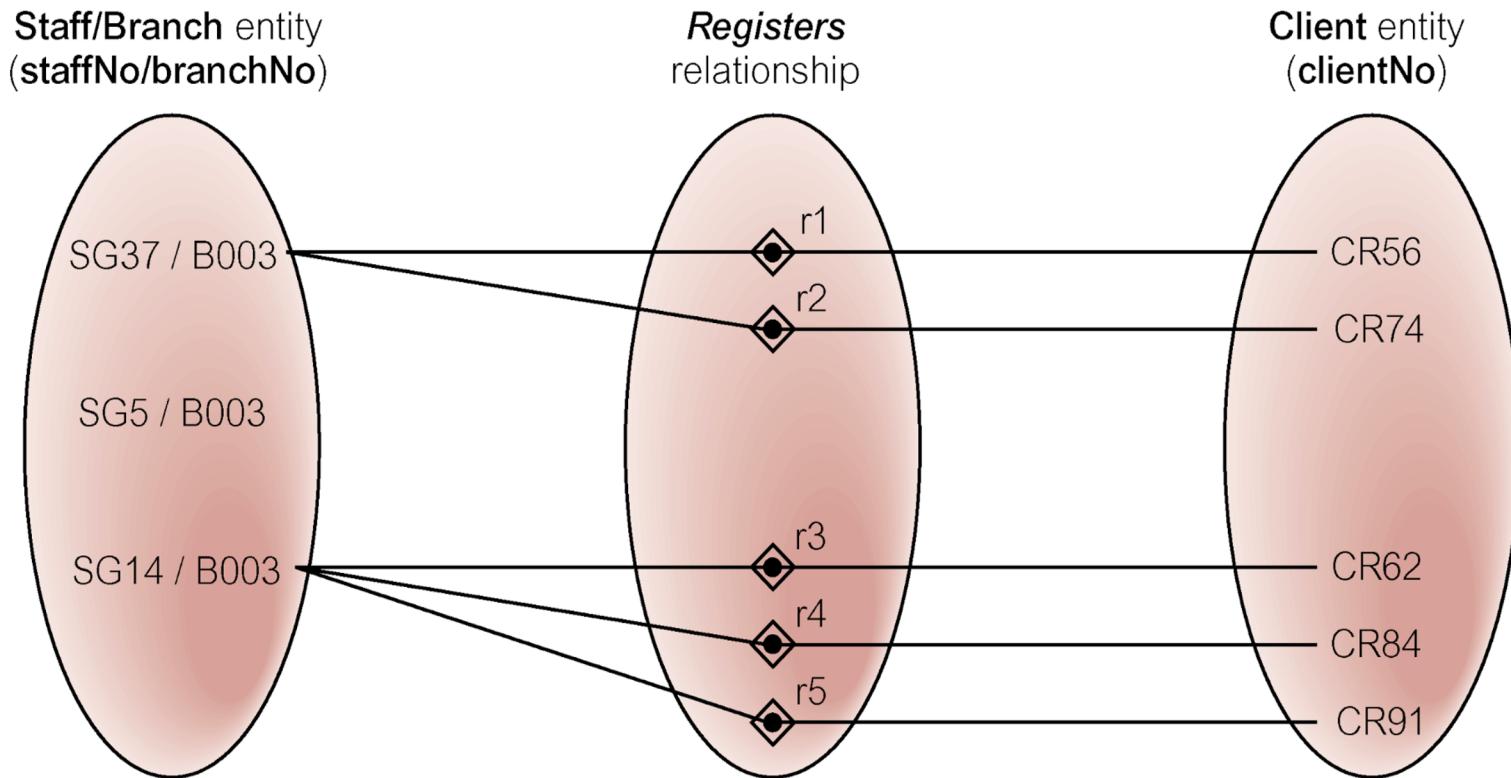
Ternary relationship called *Registers*

Diamond represents a relationship with 3 or more entity types



Semantic net of ternary *Registers*

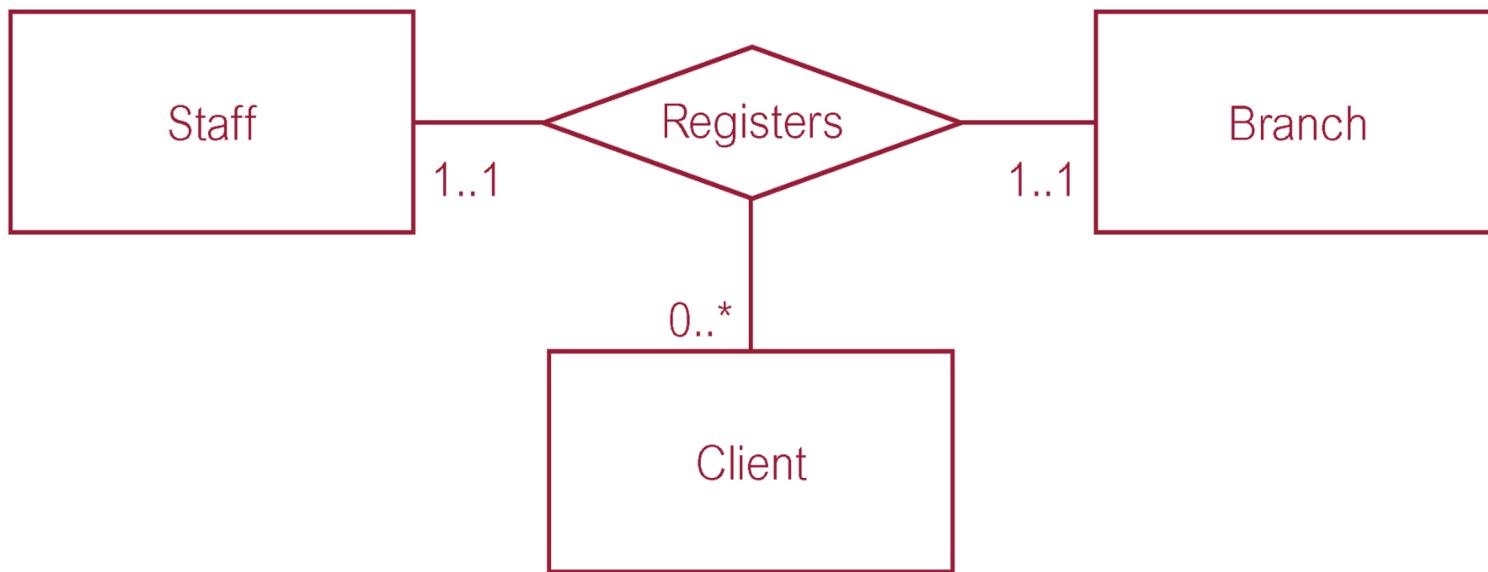
Holding the values for Staff and Branch entities fixed



Ternary relationship representation

Need to hold 2 ($n-1$) entities constant to determine multiplicity for the other entity

Example: One staff at one branch can register 0 to many clients



Ways to represent multiplicity in UML

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

Practice problems

Every sale may be associated with many sales reps, but must be associated with at least one, a sales rep can be involved with 1 or more sales

A course is attended by 1 to many students, a student may attend 0 to many courses.

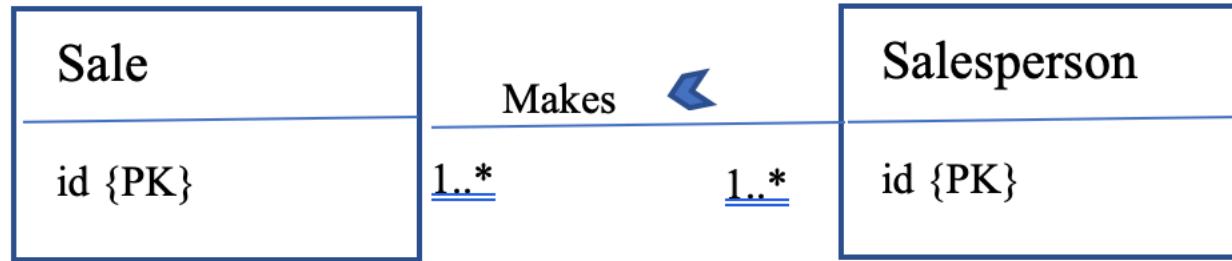
A plant is managed by only one member of a staff, all plants have one plant manager

A tweet has one author and an author can generate many tweets

A customer may place 1 to many orders at a bakery. A bakery may fill one to many orders for a customer. A bakery may have many orders and many customers.

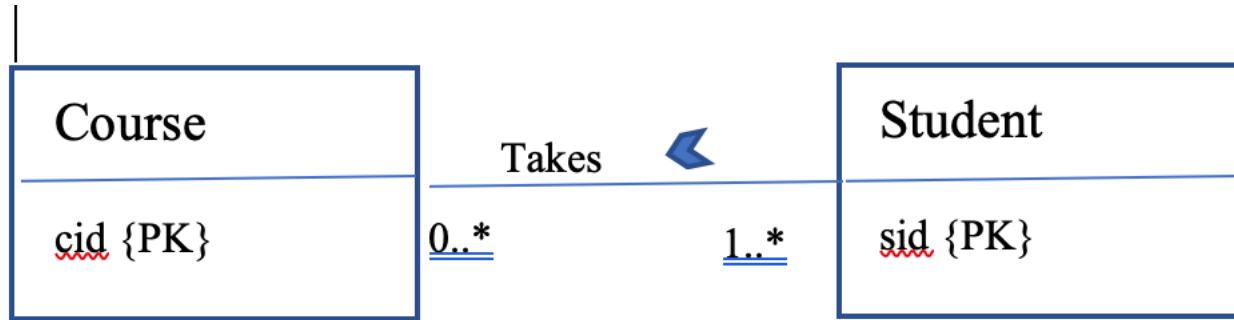
Exercise 1: create ERD using UML notation

Every sale may be associated with many sales reps, but must be associated with at least one, a sales rep can be involved with 1 or more sales



Exercise 2: create ERD using UML notation

A course is attended by 1 to many students, a student may attend 0 to many courses.



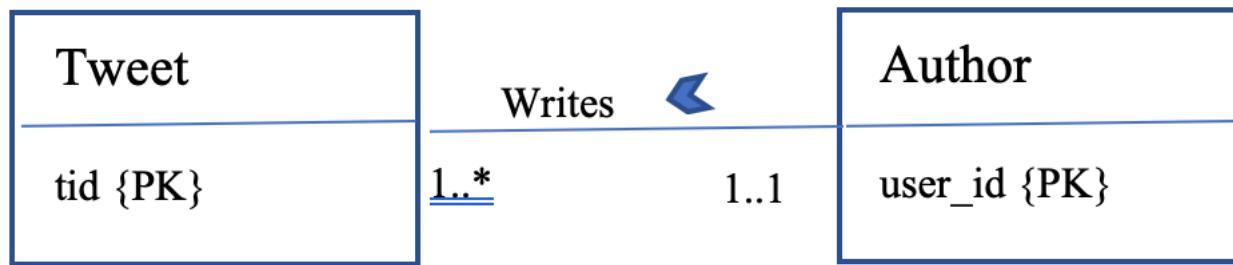
Exercise 3: create ERD using UML notation

A plant is managed by only one member of a staff, all plants have one plant manager



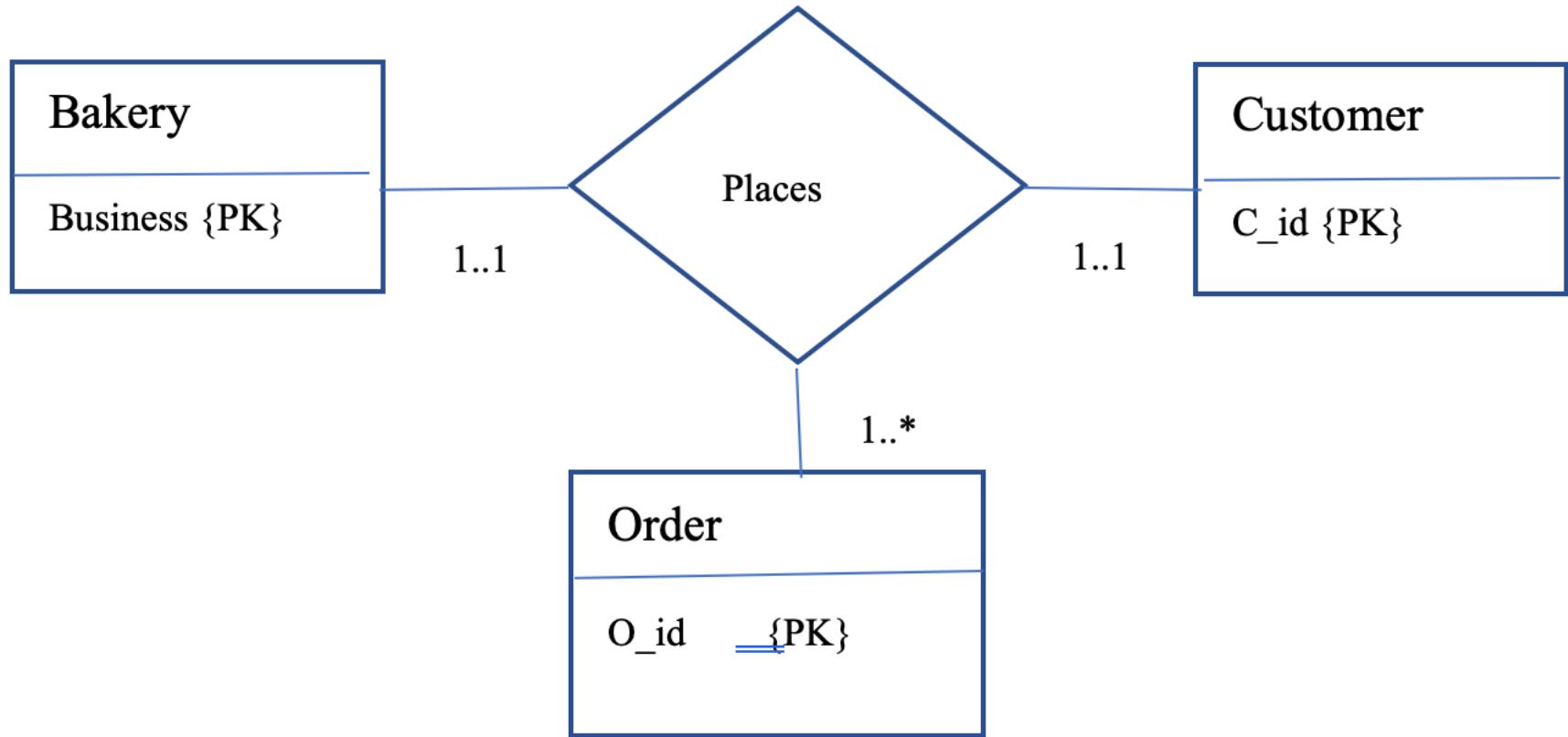
Exercise: create ERD using UML notation

A tweet has one author and an author can generate many tweets



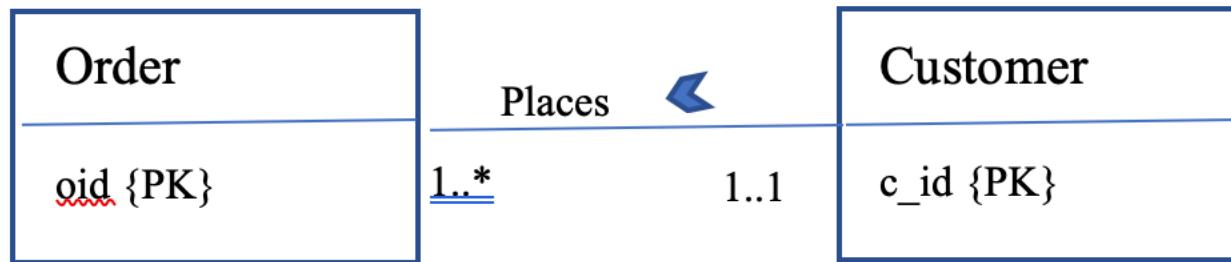
Exercise: create ERD using UML notation

A customer may place 1 to many orders at a bakery. A bakery may fill one to many orders for a customer. A bakery may have



Is there a simpler solution?

Potentially – if we really are creating a database for just one bakery



Summary

- Multiplicity allows you to express constraints on the relationships within your conceptual model
- The constraints are typically called business rules and describe the limitation of the process represented by the relationship
- Multiplicity consists of
 - Participation
 - Cardinality