



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

200
1822 - 2022

E-R Model

Man Geni Addysg Uwch yng Nghymru
Where Higher Education Began in Wales

www.uwtsd.ac.uk

Pre requisite

- Understanding of Dreamhome case study posted in Moodle

Learning Objectives

1. Introduction to E-R Models
2. ERD
3. Representation of Entities, Attributes and Relationships

E-R Data Model

- Allows us to **describe the data** involved in real world enterprise in terms of objects and their relationships
- ER – stands for **Entity-Relationship**
- Developed by **Peter Chen** in 1976
- Used in **Conceptual database design** and **modelling** phase
- Traditional approach of DB Design using DB Structures and constraints
- Very **popular modelling** technique

E-R Data Model Representation

- The model can be represented in **diagrammatic form using ER Diagrams**
- Another popular modelling method: **UML** (Unified Modelling Language)
- Used in Software Design and Creation

Principal Elements

- **Three** principal element types:

Entity Sets

- Entity-abstract object of some sort
- Entity Sets-a collection of similar entities

Attributes

- Properties of the entities in that set
- Assumptions: attributes are of primitive types, such as strings, integers, or reals

Relationships

- Connections among two or more entity sets

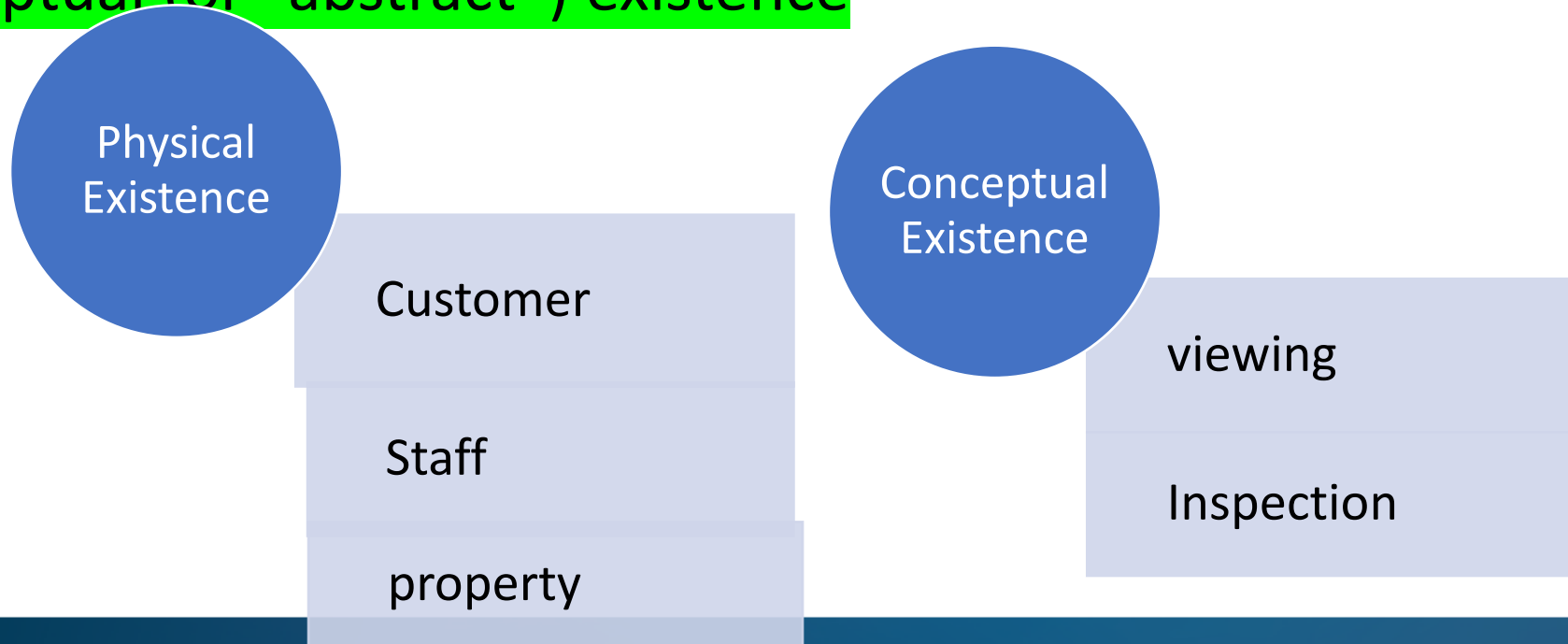
Entity & Entity Sets

Entity

- **Objects in real life** that are distinguishable from other objects.
- **Usually nouns** in a scenario description but not every noun becomes an entity – think carefully when you finalize on the entities
- E.g. If you are building a data model for a small retail store, then the potential entities may be
 - Customer
 - Team
 - Sales Rep
 - Order
 - Item
 - Inventory list

Entity Types

- A group of objects with the same properties, which are identified by the enterprise as having an independent existence.
- objects with a physical (or “real”) existence or objects with a conceptual (or “abstract”) existence



Diagrammatic representation of entity types

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



Staff

Branch

Types of Entities

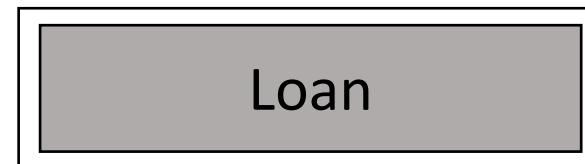
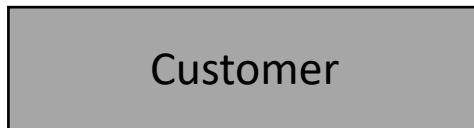
Regular Entities

- Called as Strong Entities.
- Independent Entities
- E.g. customer

Weak Entities

- Its existence depends on strong entity
- E.g. Loan
 - If an customer Instance is deleted, then the dependants too are deleted. They do not exist without the customer.

Notation for a Regular Entity



Entity Sets

- **Collection of similar entities.**
- The entity set need **not be disjoint**.
- For e.g. an employee can belong to two different departments, a student can be part of more than one module etc.
- In programming parlance,
an entity set = type definition and an entity = variable of that type

Movie-Database example:

Each Movie = Entity
Set of all movies = Entity Set

Attributes

Attribute Types

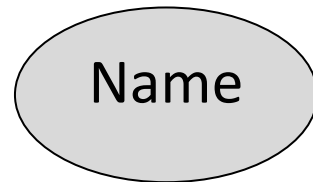
- Each entity has attributes – **properties that describe it.**
- A particular entity will have values for its attributes.
- E.g. an employee entity may have name, age, home address, home phone etc as their attributes.

Attribute Types			
Simple vs Composite	Single vs Multivalued	Stored vs Derived	Null vs Complex

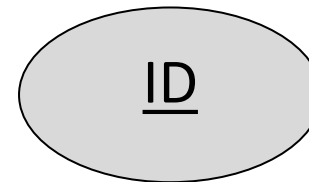
Attributes Notation

- **Attribute Domain:** The set of allowable values for one or more attributes.
- Each attribute is associated with a set of values called a domain.
- **Notation:** Attributes are represented by ovals.

Notation for an Attribute



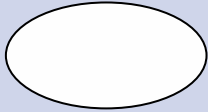
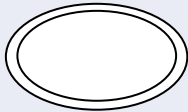
Notation for an **Key** Attribute



Attribute Types - 1

Type	Description	Example
Simple Attribute	An attribute composed of a single component with an independent existence. Also called as atomic attributes.	Employee entity: Position, salary etc.
Composite Attribute	An attribute composed of multiple components, each with an independent existence.	Employee entity: Address → can be made up of <i>street, town, country, postcode</i>

Attribute Types - 2

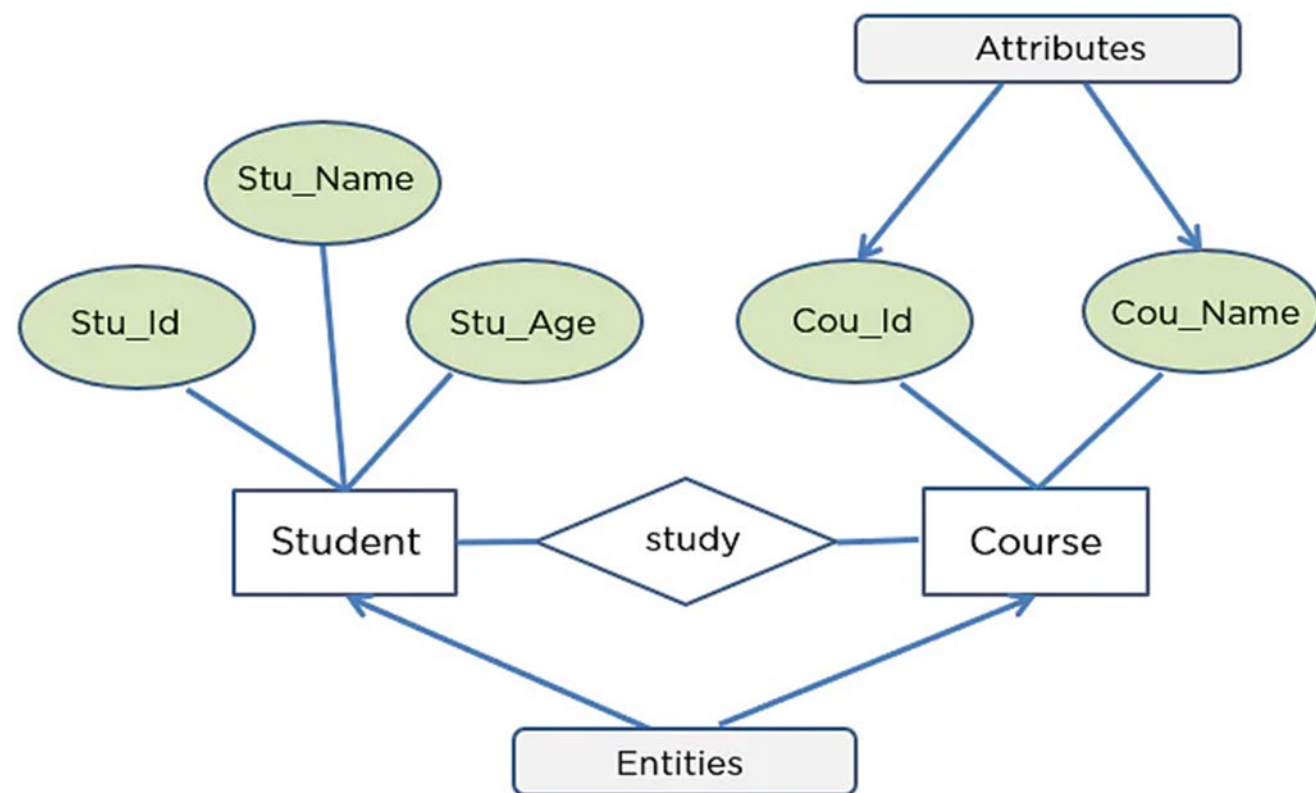
Type	Description	Example
Single Valued Attribute 	An attribute that holds a single value for each occurrence of an entity type.	Employee entity: Salary → can be only one value. Cannot draw more than one salary.
Multi Valued Attribute 	An attribute that holds multiple values for each occurrence of an entity type.	Employee entity: PhoneNo → A person can have one or more mobile phone number, landline etc.

Attribute Types - 3

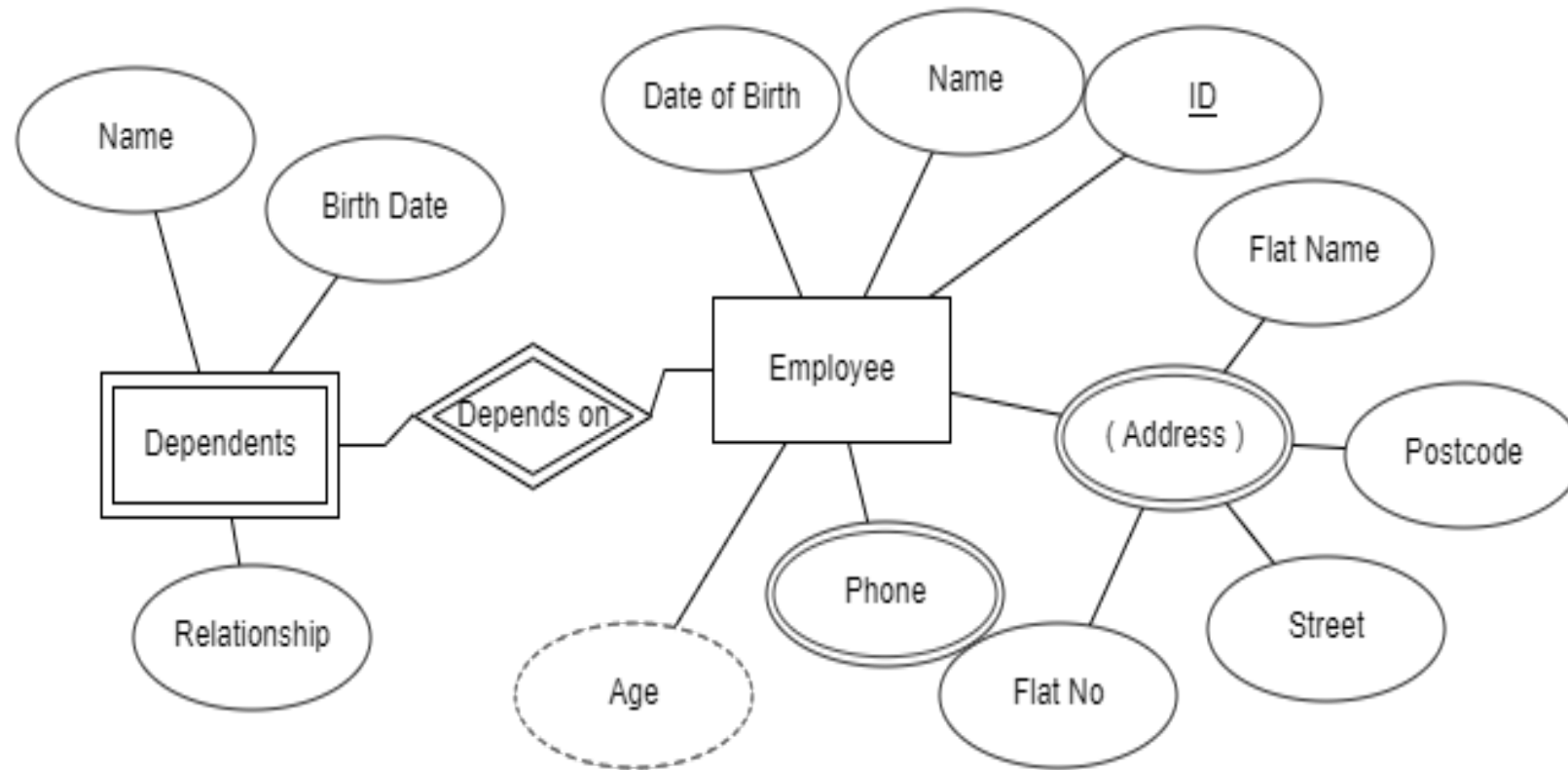
Type	Description	Example
Stored Attribute	An attribute that holds a value provided as is.	Employee entity: Name → can be set by the user as used as is.
Derived Attribute 	An attribute that represents a value that is derivable from the value of a related attribute or set of attributes, not necessarily in the same entity type.	Employee entity: Age → can be derived from dateOfBirth and currentDate.

Attribute Types - 4

Type	Description	Example
NULL Attribute	An attribute that means either the value is unknown or inapplicable. It does not mean zero or blank.	Employee entity: Commission → can be NULL
Complex Attribute 	An attribute that is formed by nesting composite attributes and multi-valued attributes in arbitrary way. We can say this as the both are in the attribute.	Employee entity: A person can have many phone numbers, many e-mail addresses, home addresses etc. So if there is an attribute in the name of 'Contact detail', it can be a complex attribute.



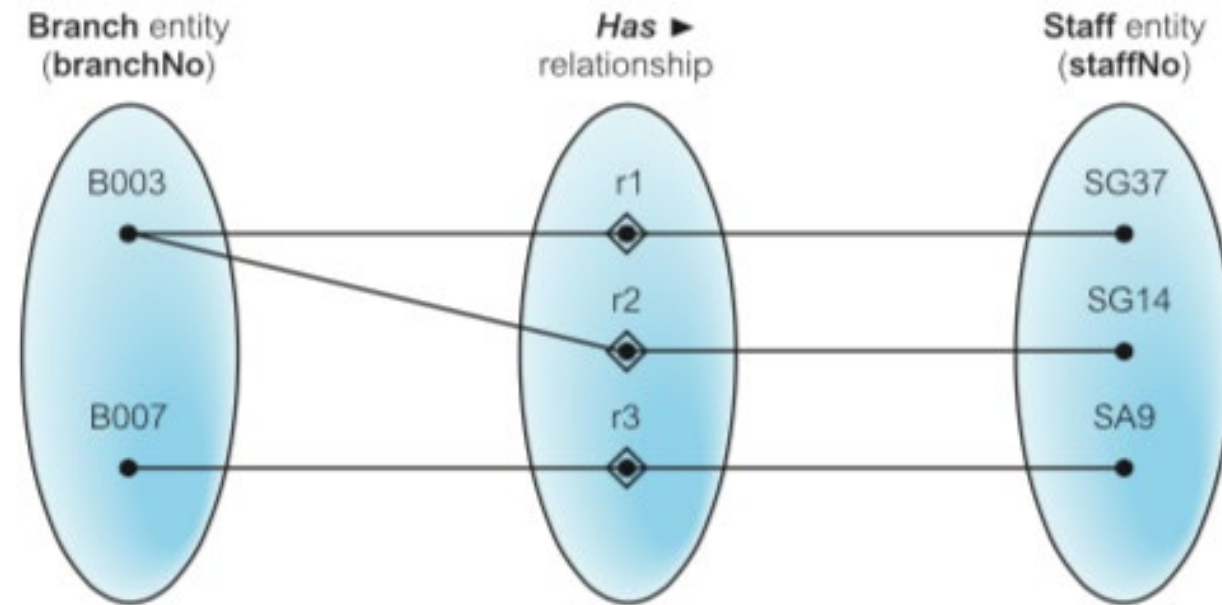
Entity & Attributes



Relationships

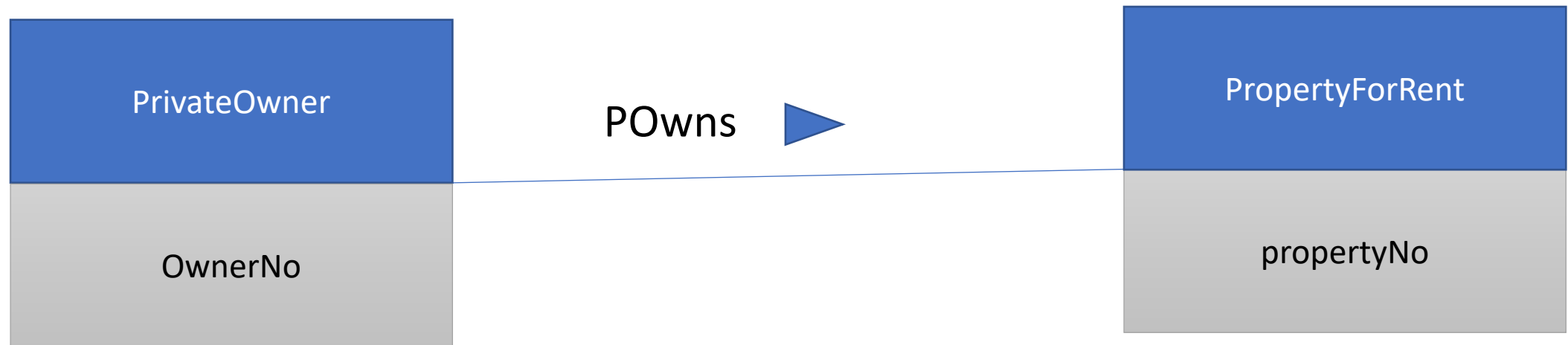
Relationship & Relationship Sets

- Relationship → An **association among the entities**.
- Entities involved in a relationship are said to be the **participants** in that relationship.



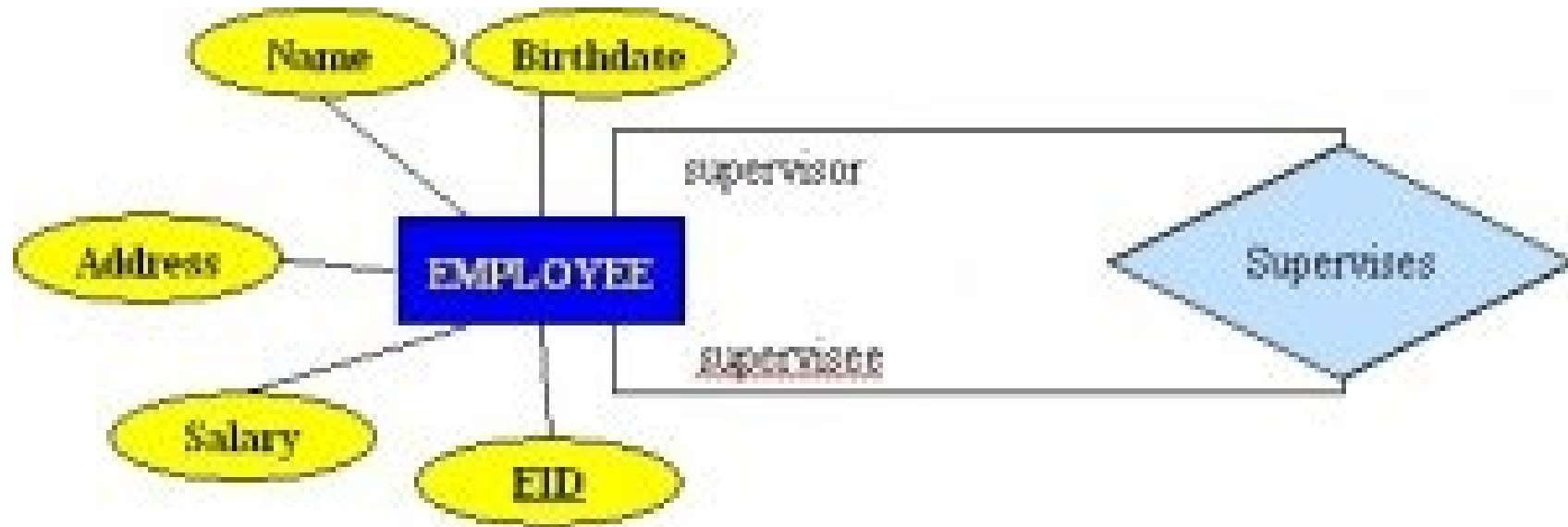
Relationship Types

- meaningful associations among entity types.



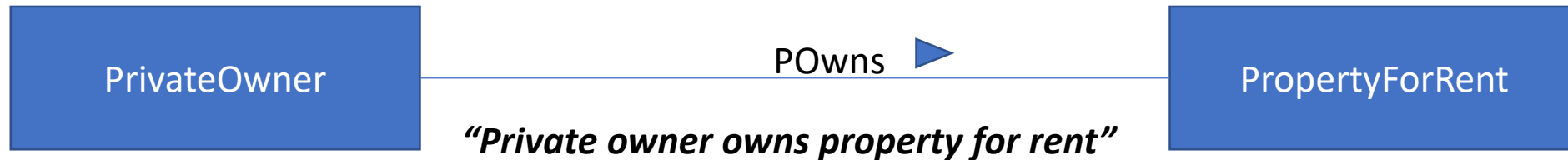
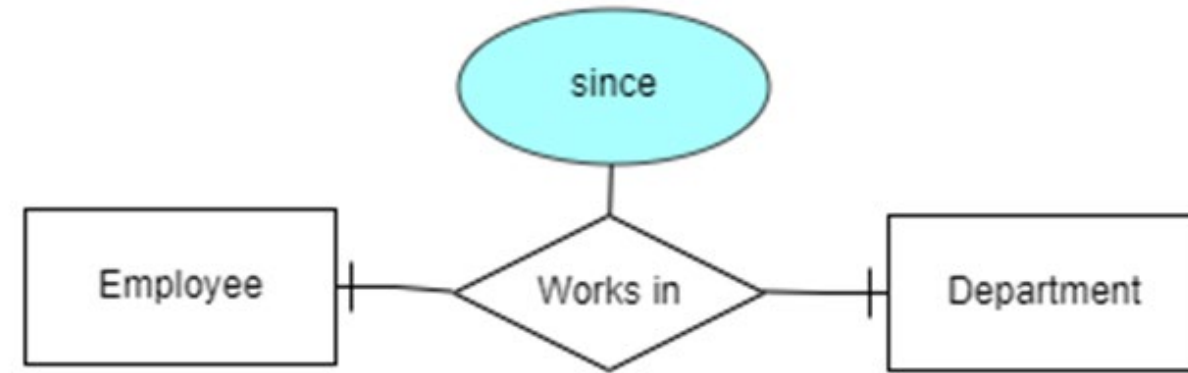
Unary Relationship - Example

- Also called as **recursive** relationship.
- Relationship exists between occurrences of the same entity set.



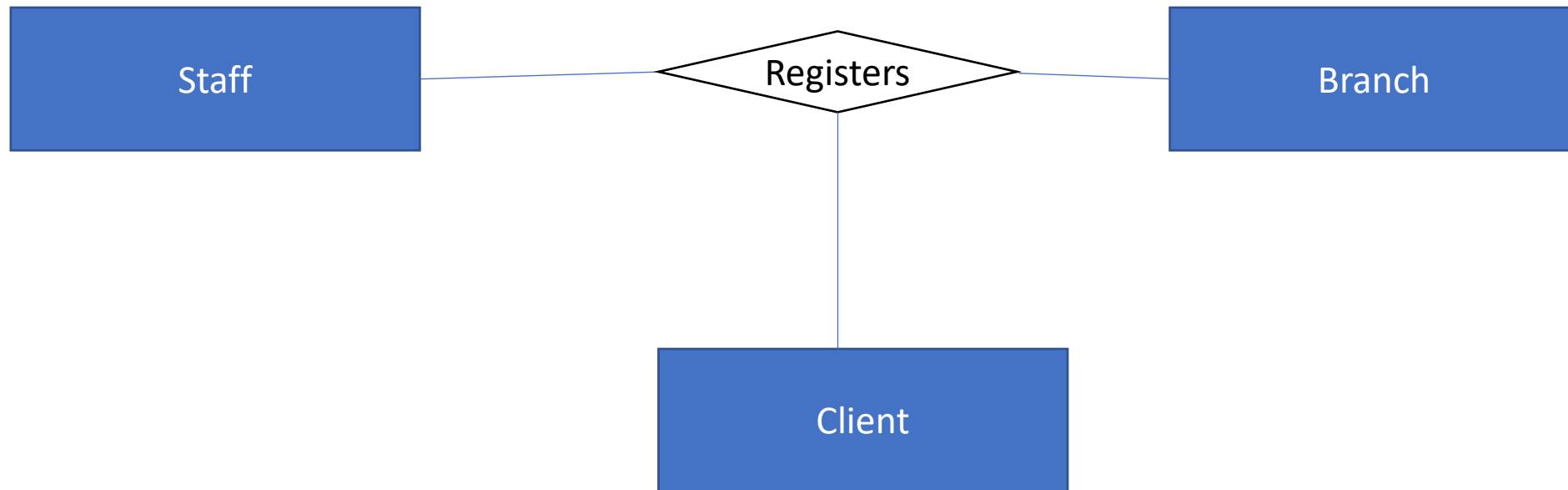
- Binary Relationship

- A relationship of degree two is called **binary**

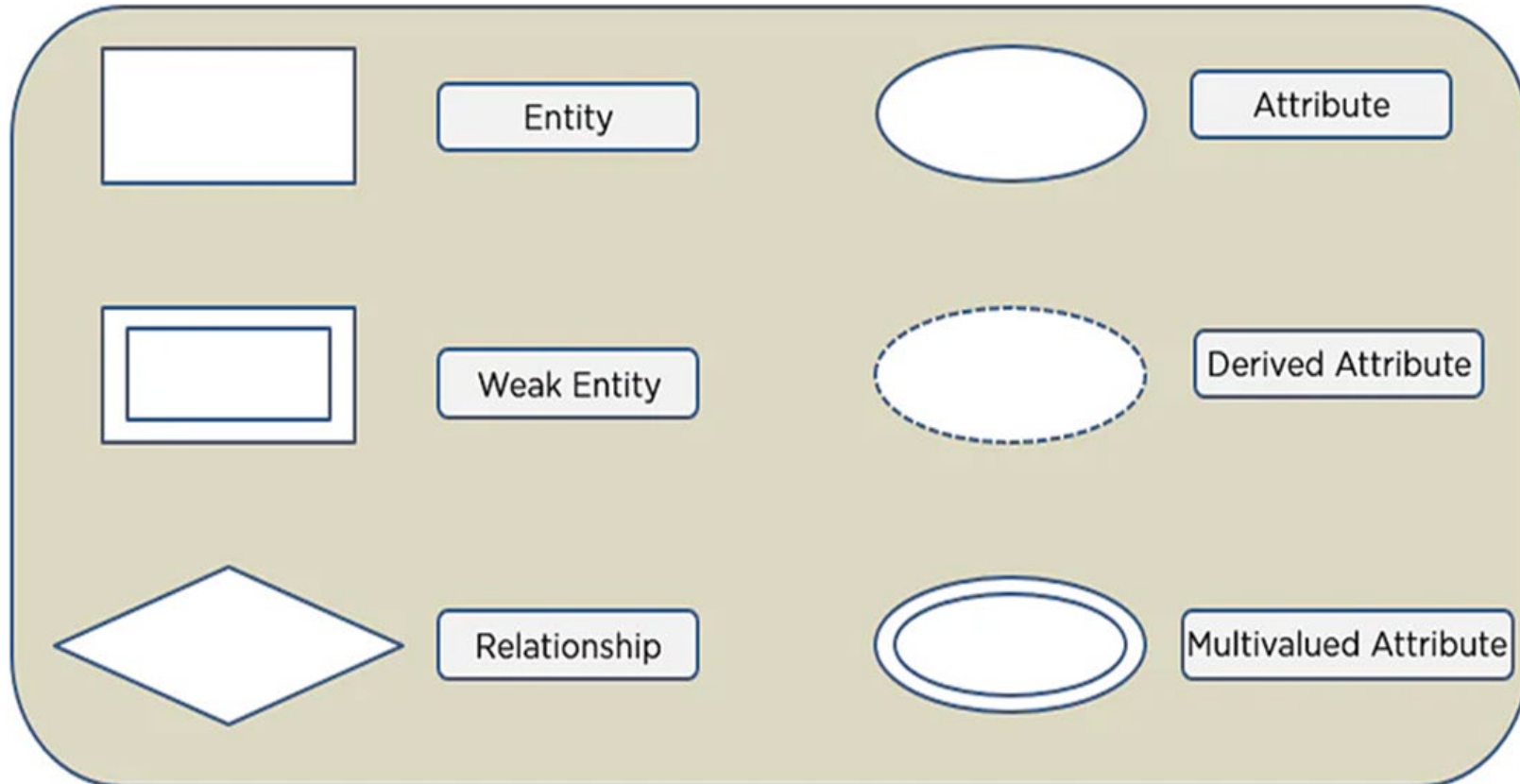


- Ternary Relationship

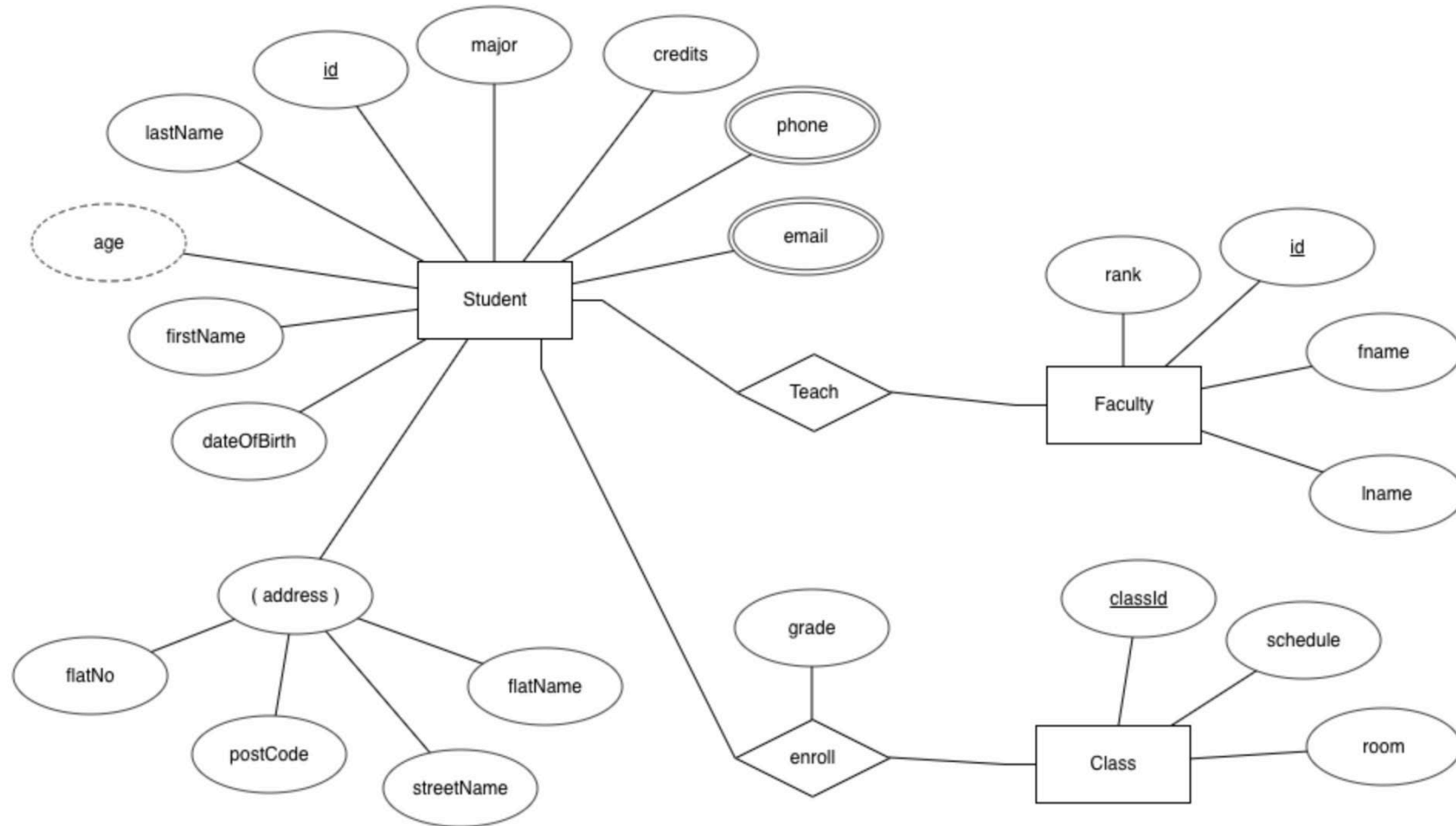
- A relationship of degree three is called ternary



E-R Diagram Notations



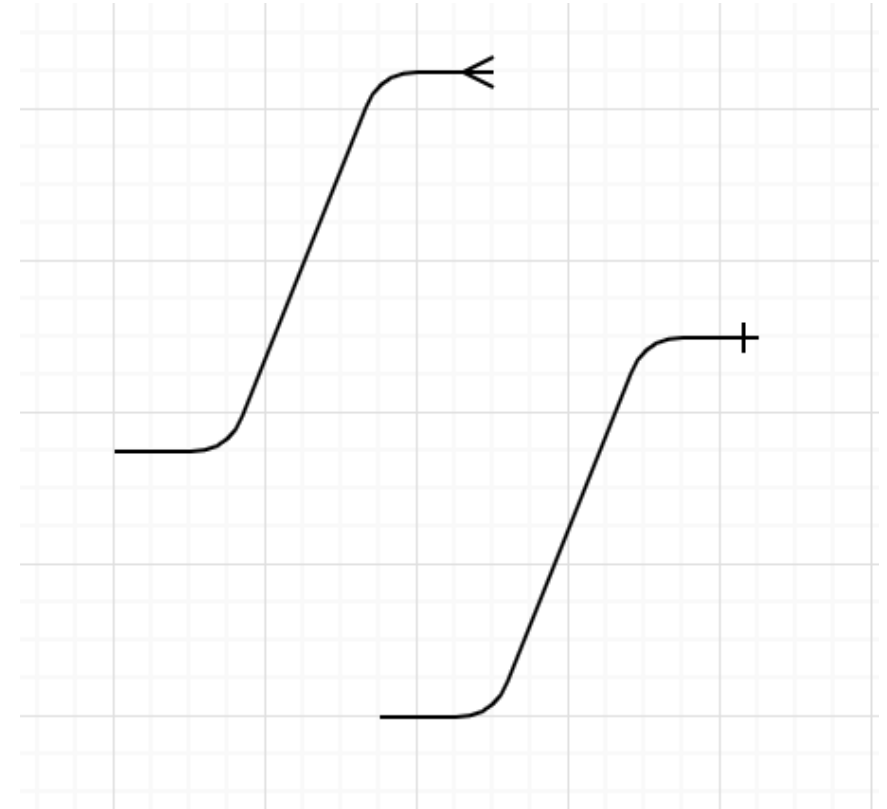
University Database – Relationship Example



Relationship Cardinality

Cardinality of Binary Relationships – 4 Types

- 1:1 (one to one), 1:n (one to many), n:1 (many to one), n:n (many to many)
- A many to one relationship is a special case of many to many relationship.
- A one to one relationship is a special case of a many to one relationship.
- Use **Crow's foot** notation to represent the cardinality, using a line for 1 and 3 toes (resembling crow's feet) for many.



1:1

- Relationship where each entity instance in X is associated with at most one entity instance in Y and, conversely, each entity instance in Y is associated with at most one entity instance in X.
- E.g. a member of staff manages a branch

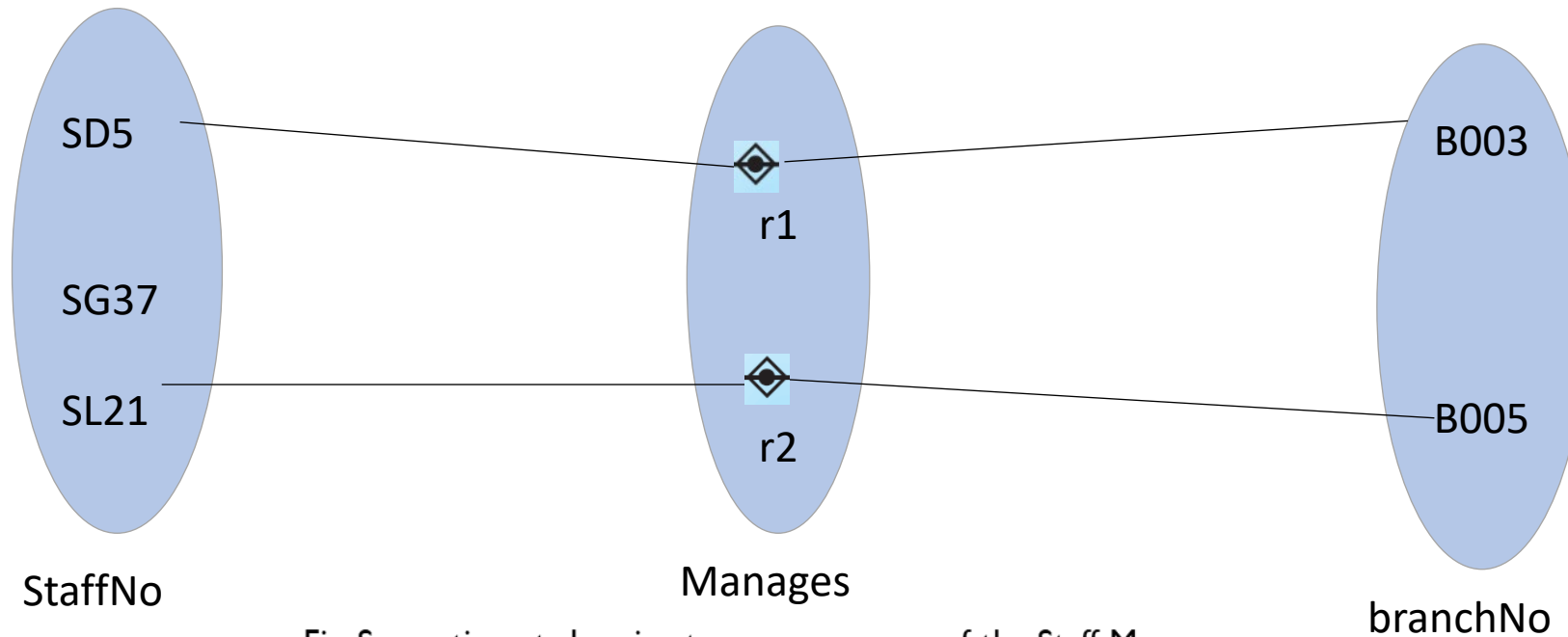
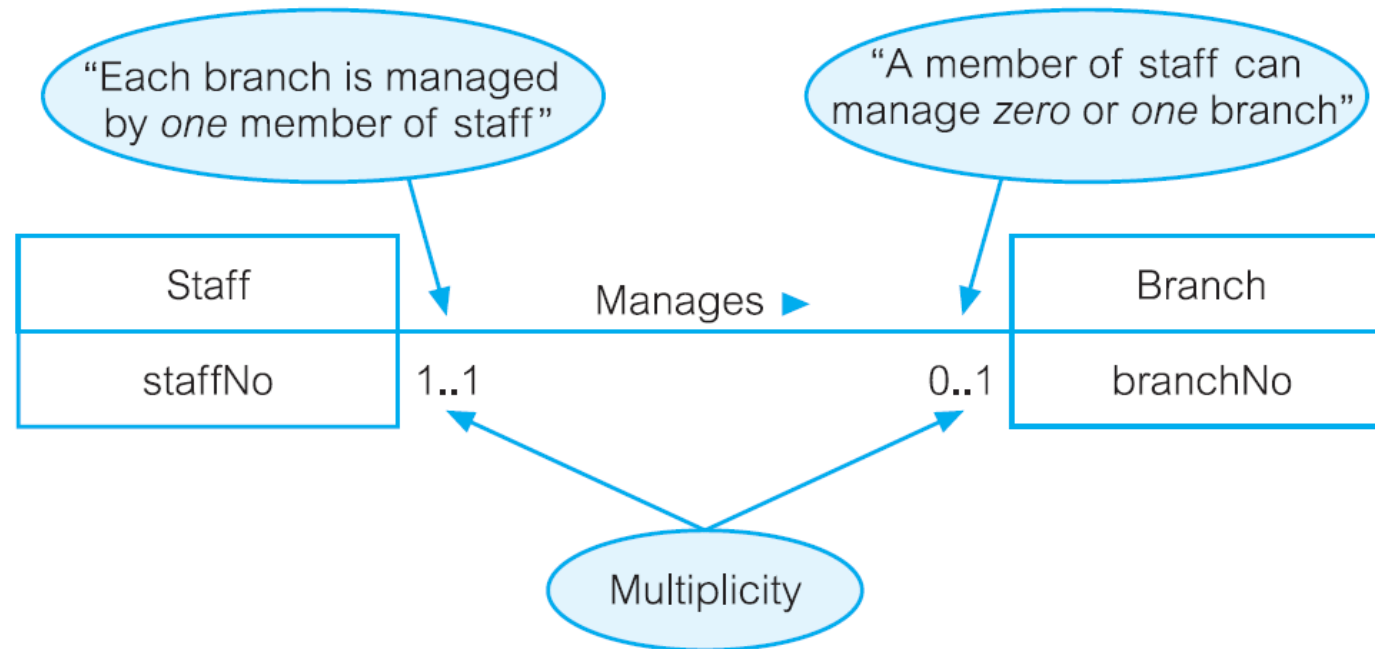


Fig: Semantic net showing two occurrences of the Staff Manages

- Determining the multiplicity

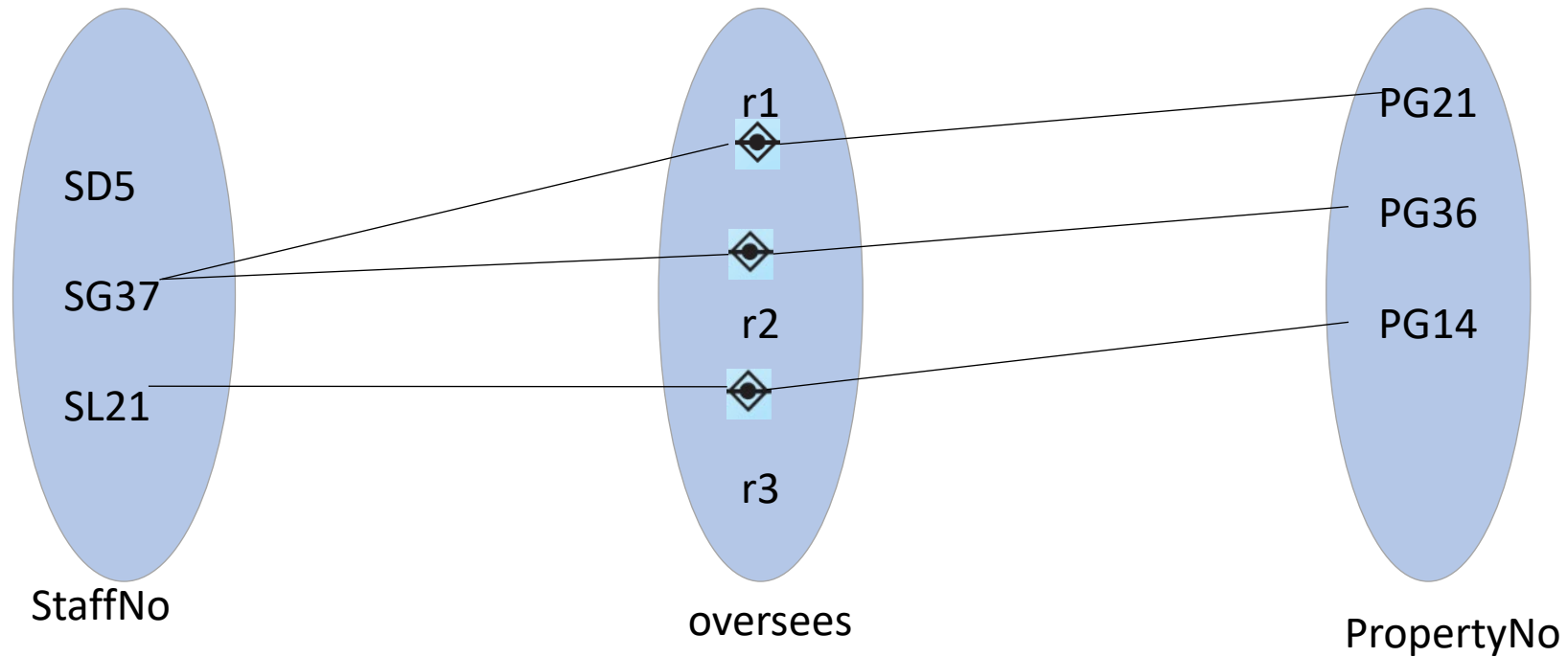


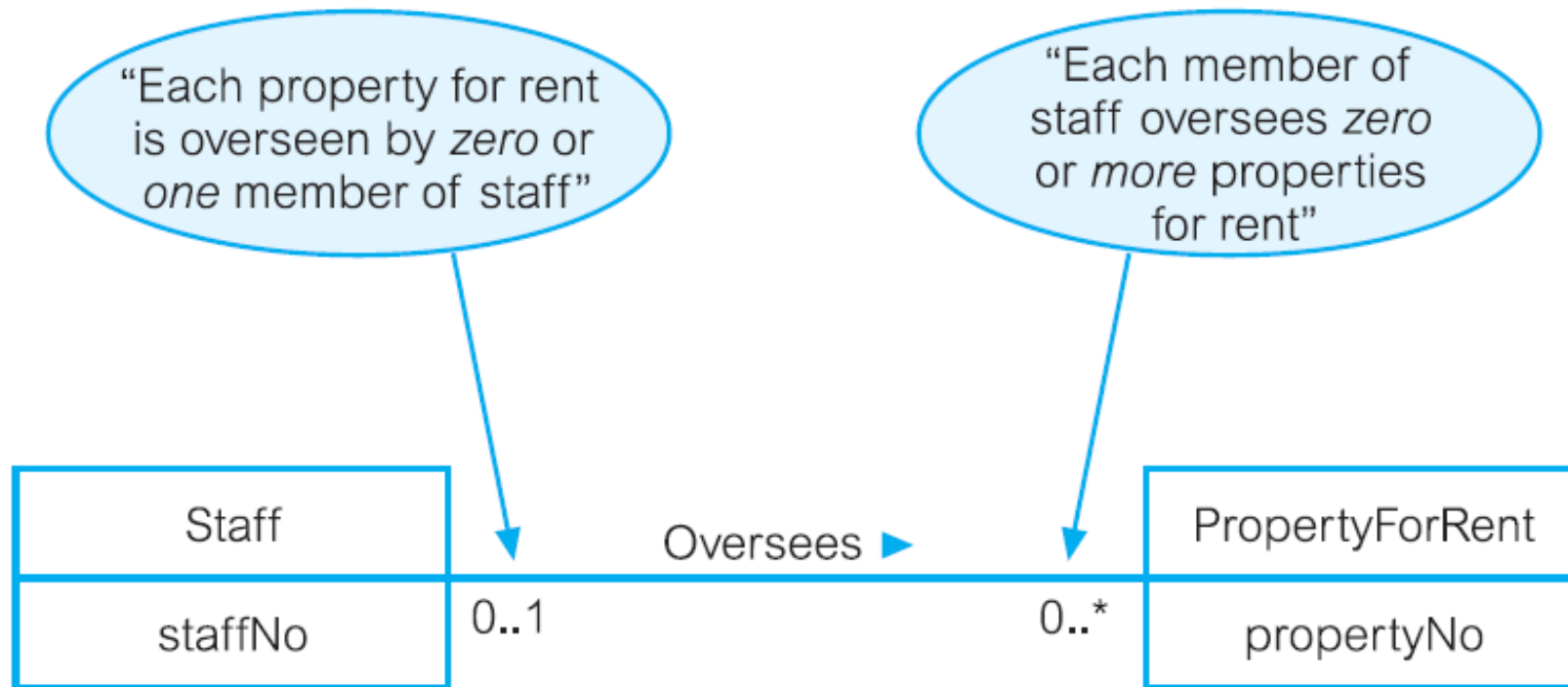
Connolly, T., & Begg, C. (2005). Database systems: a practical approach to design, implementation, and management. Boston, MA: Pearson Addison-Wesley.

1:n

- Relationship where each entity instance in X can be associated with many entity instances in Y but each entity instance in Y is associated with at most one entity instance in X .
- E.g. a member of staff oversees properties for rent (1:*);

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

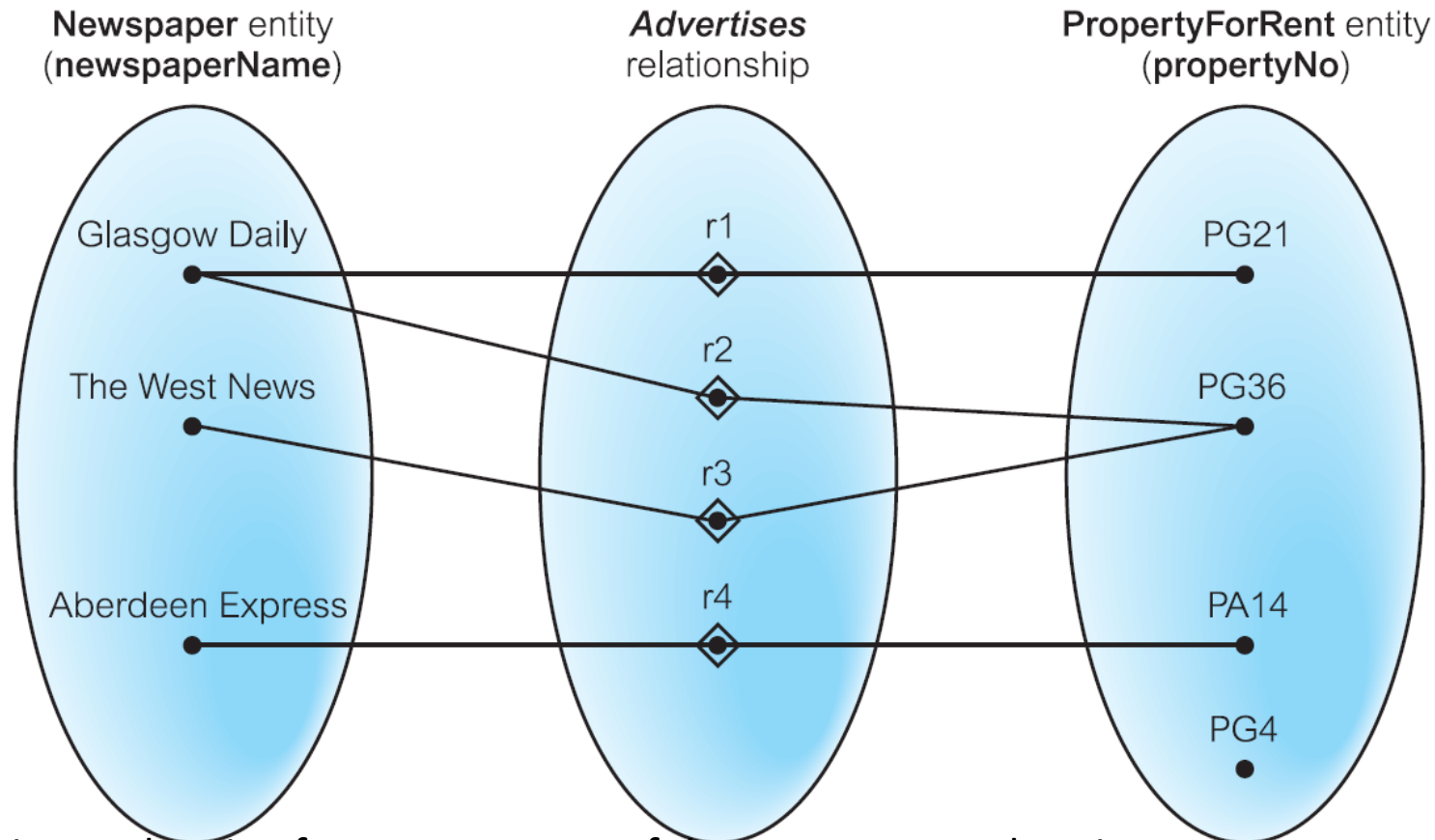




n:n

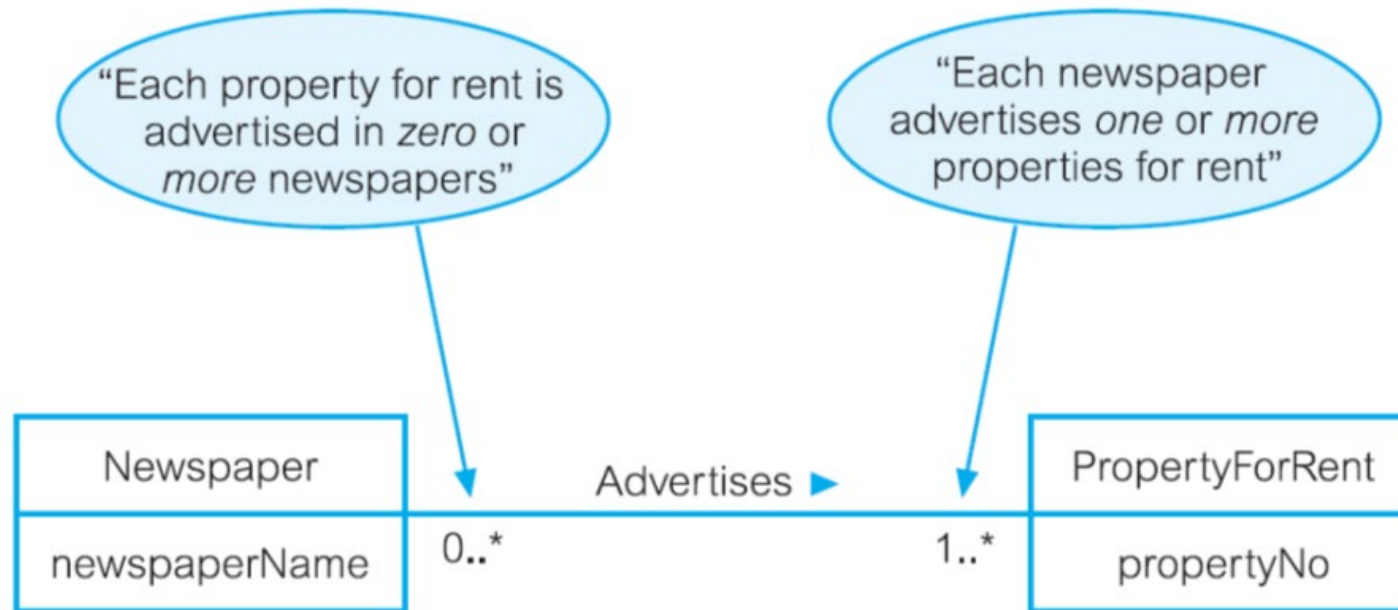
- Relationship where each entity instance in X can be associated with many entity instances in Y and each entity instance in Y can be associated with many entity instances in X .
- E.g. newspapers advertise properties for rent (*:*)).

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



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

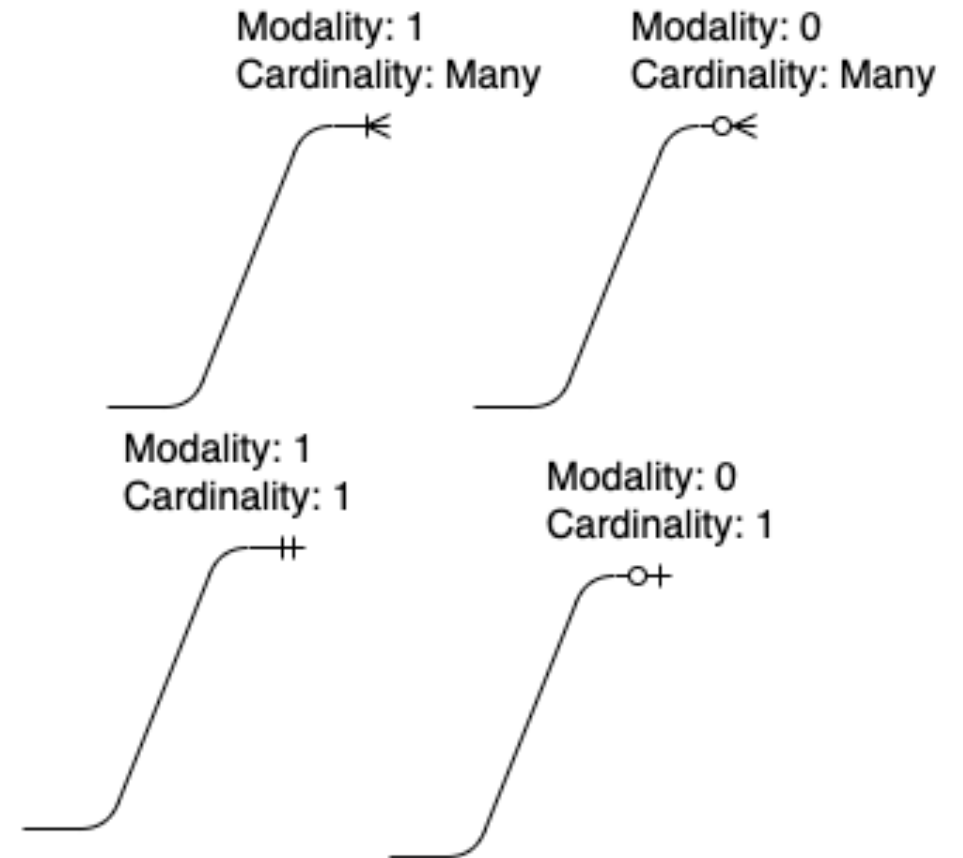
Connolly, T., & Begg, C. (2005). Database systems: a practical approach to design, implementation, and management. Boston, MA: Pearson Addison-Wesley.



Modality

Modality in a Relationship

- Refers to the minimum number of times an instance in one entity can be associated with an instance in the related entity.
- Modality can be **1 (line)** or **0 (circle)** and the symbol is placed on the inside, next to the cardinality symbol.



Summary

a. ERD Elements

i. Entities

- i. Weak,
- ii. Strong

ii. Attributes

- i. Simple vs Composite
- ii. Single Value vs Multi value
- iii. Stored vs Derived
- iv. Null vs Complex

iii. Relationships - Degree

- i. 1-ary, Bi-nary, N-ary,