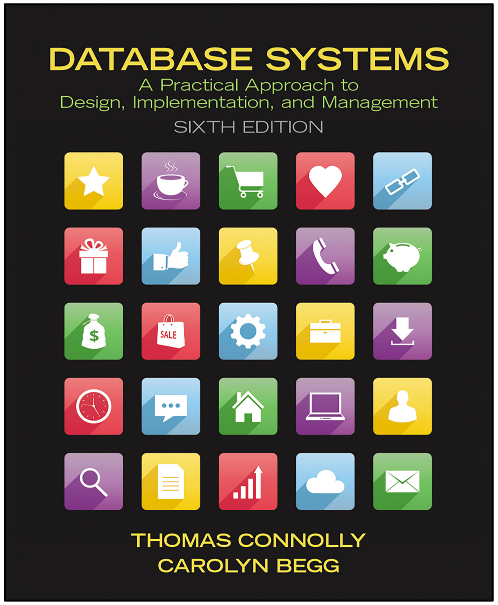
Conceptual modeling

Topic 1

Lesson 1 – Entity Relationship model with UML notation

Chapter 12 12.1-12.5 Connolly and Begg



Conceptual model

Goal: a model that is nontechnical and is free of ambiguities: that represents the data objects and relationships of the systems

Process: it is a **top-down approach** where we identify the important data called entities and the relationships among them. We then identify the information we want to hold on the entities and the relationships (called attributes).

What is a model?

UML notation for conceptual modeling

UML is a good notation for conceptual data modeling because the representation stands apart from implementation choices specific to the database solution.

The same conceptual design can be used to build a relational database, an object-oriented database, a simple data store, or some NoSQL data model

Basic objects: entity and a relationship

A square represents an **entity.**

Entity type

Group of objects with same properties, identified by enterprise as having an independent existence.

Entity occurrence

Uniquely identifiable object instance of an entity type.

A line represents a binary **relationship** and connect entity instances

Relationship type

Set of meaningful associations among entity types.

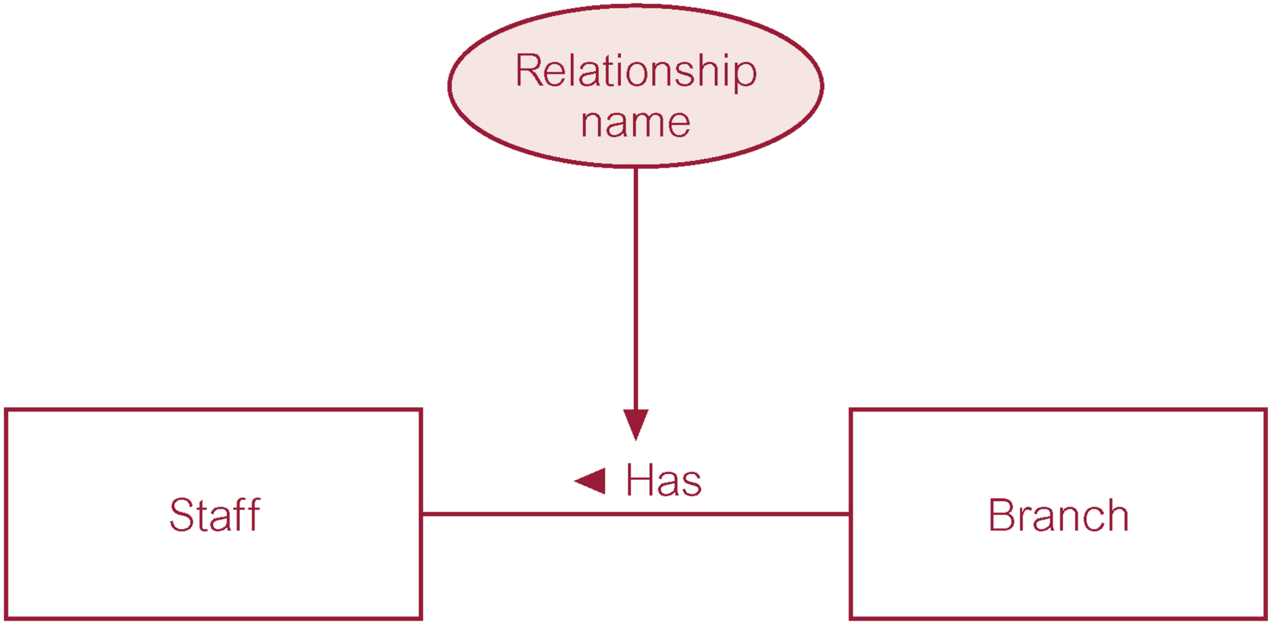
Relationship occurrence

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

ER diagram of Branch *Has* Staff relationship

A relationship is only labeled in one direction

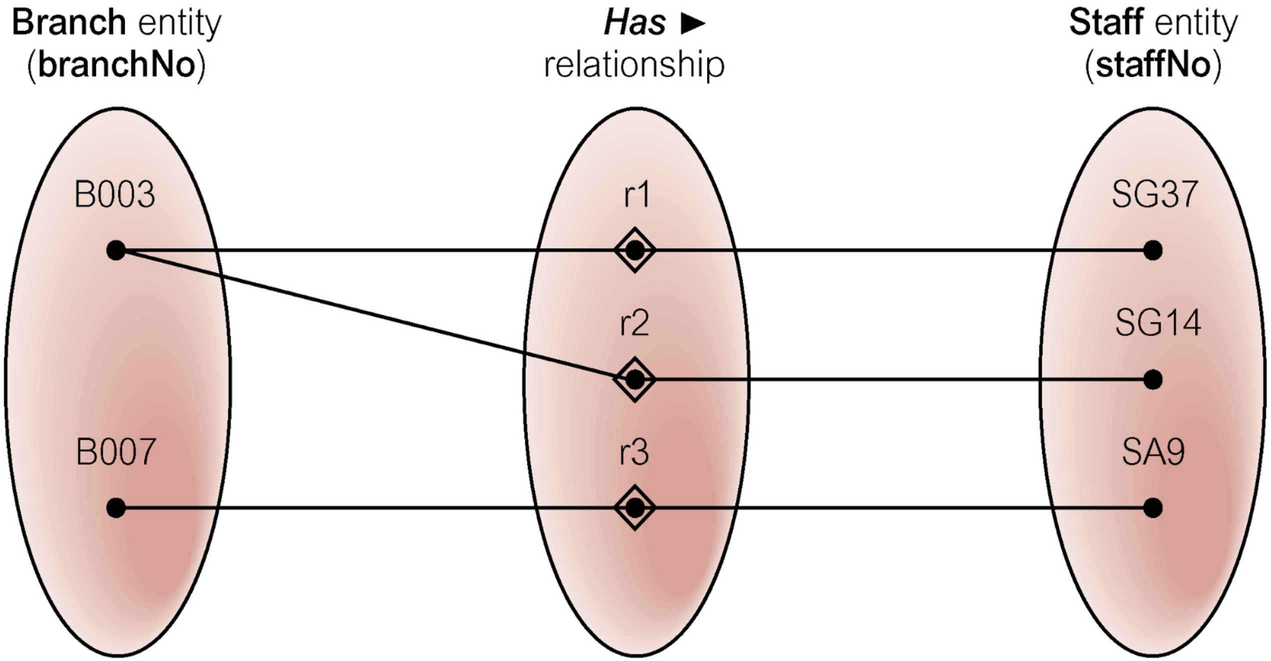
You need to be able to name the relationship in both directions



'Branch has staff'

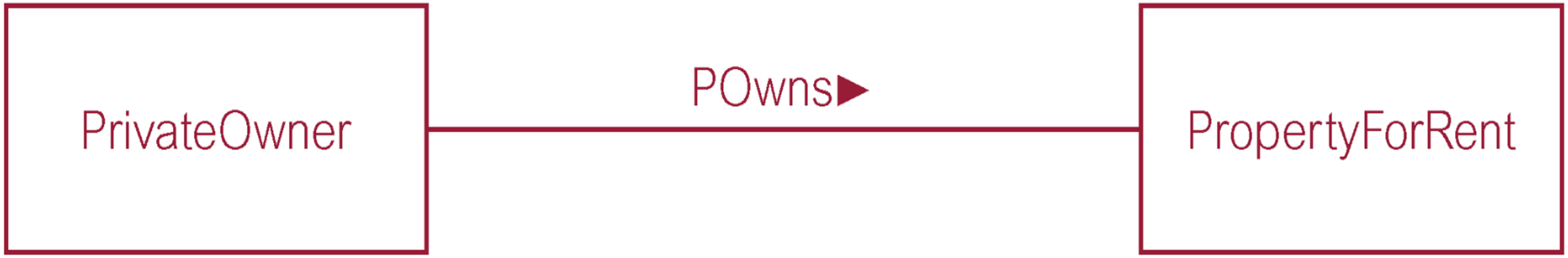
Semantic net of *Has* relationship type

- A detailed object-level model
- A dot represents an entity occurrence
- A diamond represents a relationship occurrence



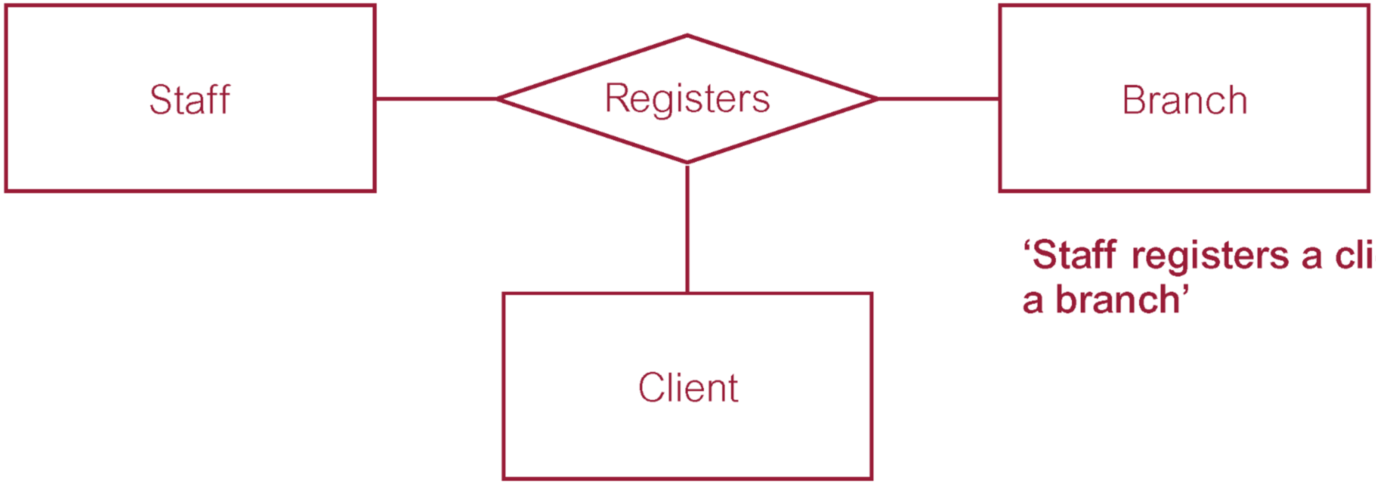
Binary relationship called *POwns*

‘Private owner owns property for rent’



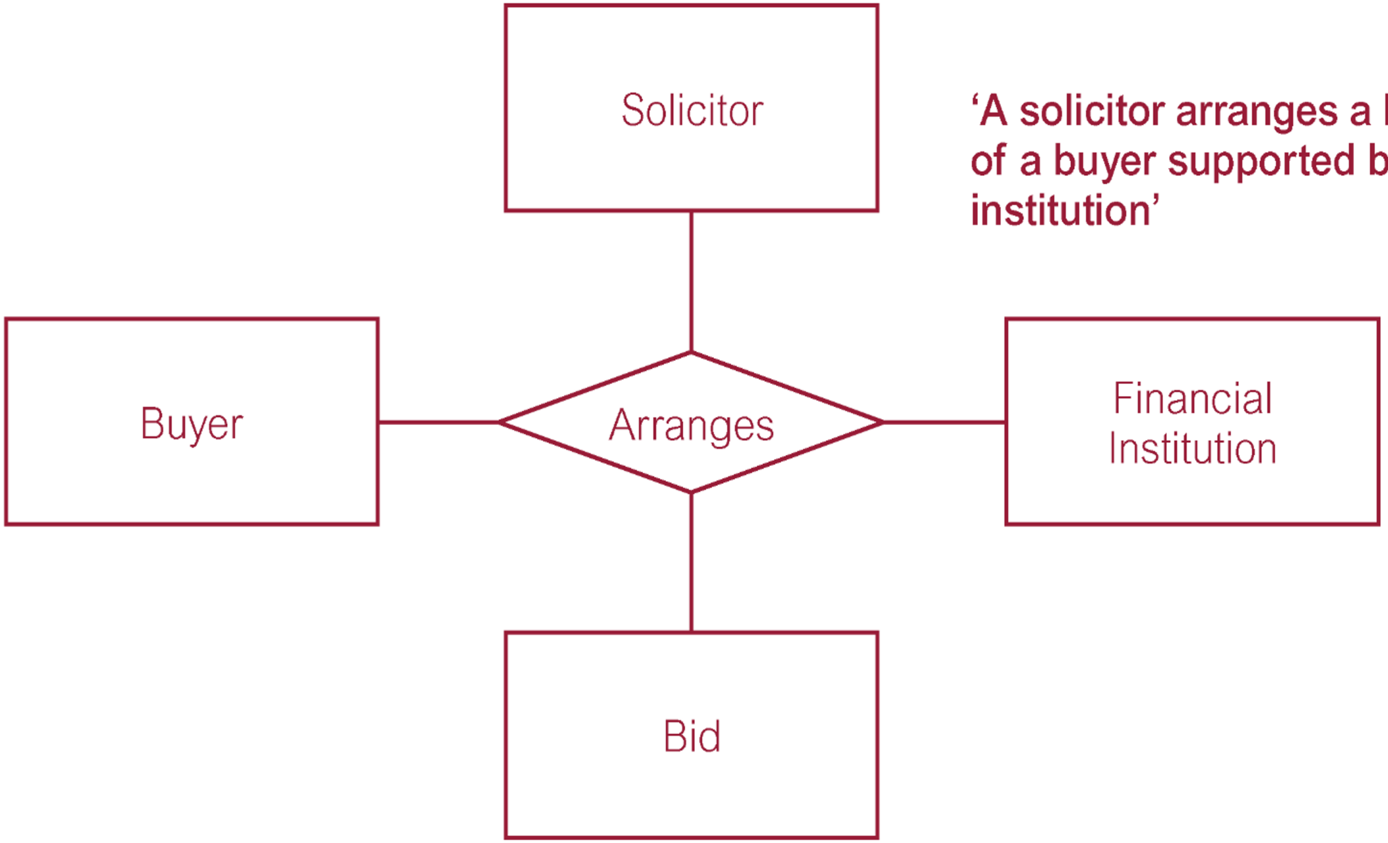
Ternary relationship called *Registers*

Diamond represents a relationship with 3 or more entity types



'Staff registers a client at a branch'

Quaternary relationship called *Arranges*



'A solicitor arranges a bid on behalf of a buyer supported by a financial institution'

How many entities in a relationship?

We have seen relationships consisting of 2, 3, and 4 entity types.

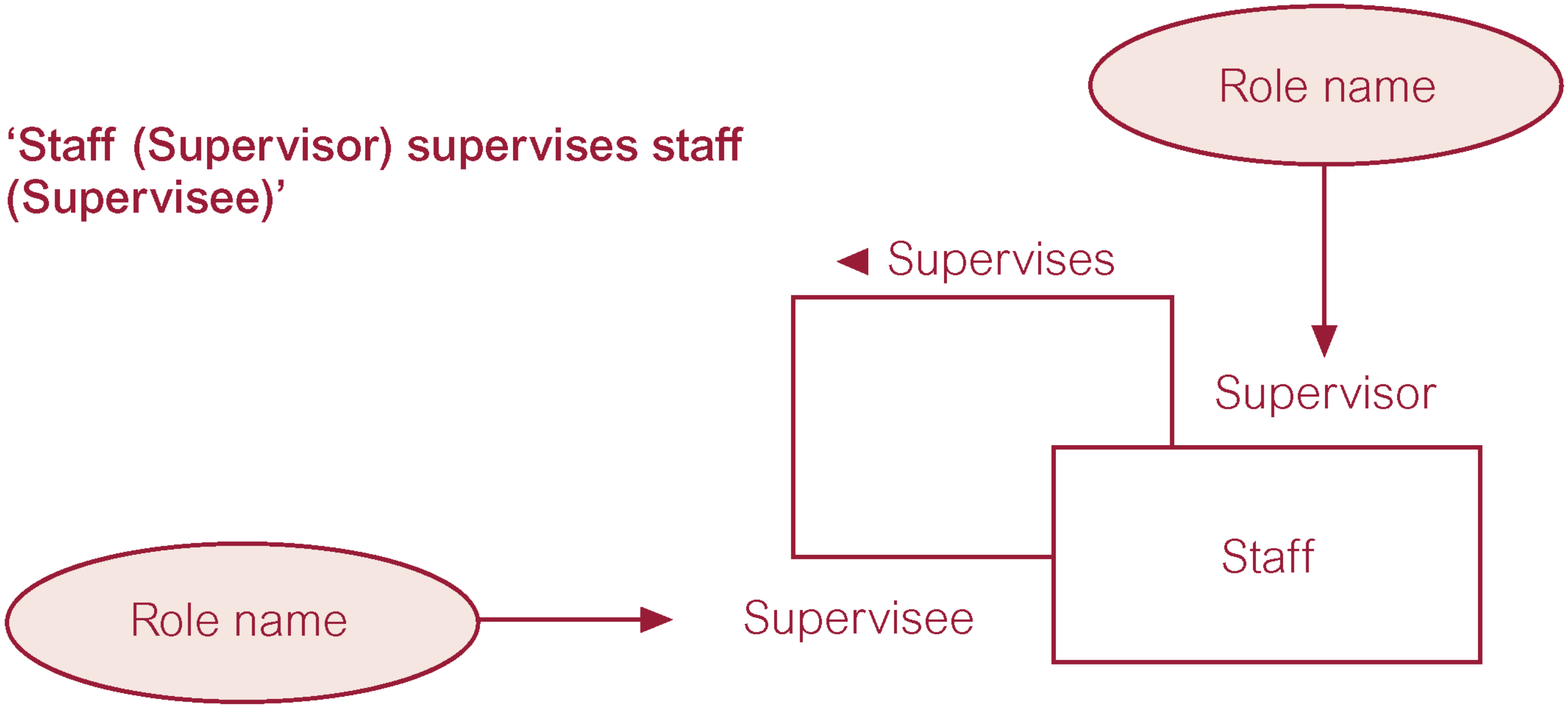
Does a relationship always have to have >= 2 entity types?

Can we have a relationship with 0 entities?

Can we have a relationship with only 1 entity?

Recursive relationship

Involves only 1 entity type.



‘Staff (Supervisor) supervises staff (Supervisee)’

Attribute

An attributes is a property of an entity or a relationship type.

Attributes hold values that describe each entity occurrence.

An attribute domain: the set of allowable values for one or more attributes.

There are many different types of attributes.

Characteristics of entity types

Simple Attribute

Attribute composed of a single component with an independent existence.

Composite Attribute

Attribute composed of multiple components, each with an independent existence.

Single-valued Attribute

Attribute that holds a single value for each occurrence of an entity type.

Multi-valued Attribute

Attribute that holds multiple values for each occurrence of an entity type.

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 type.

Identifying entity instances

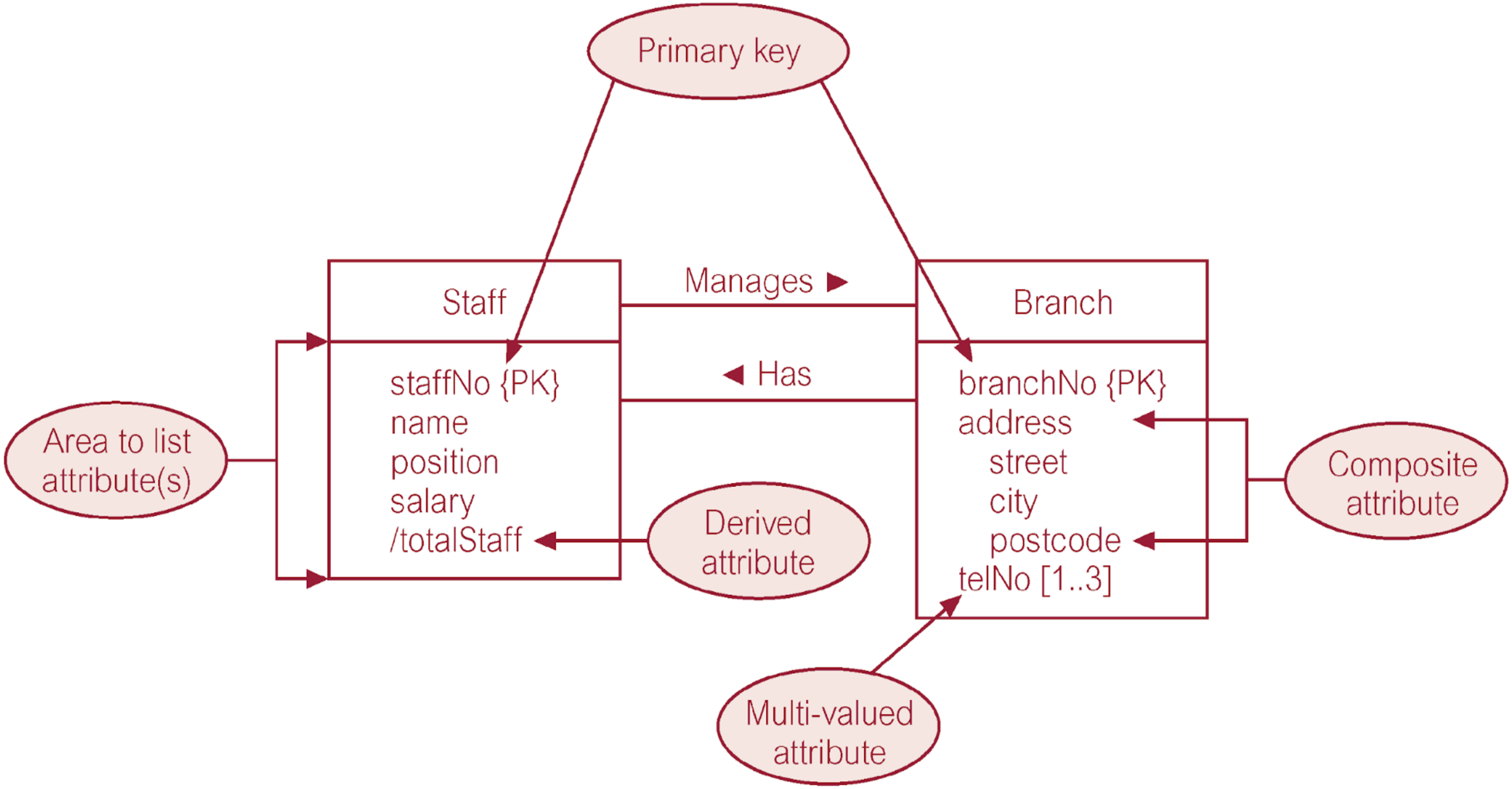
Attributes can be declared as PRIMARY **KEY {PK}**

The {PK} is the chosen candidate key that uniquely identifies each occurrence of an entity type.

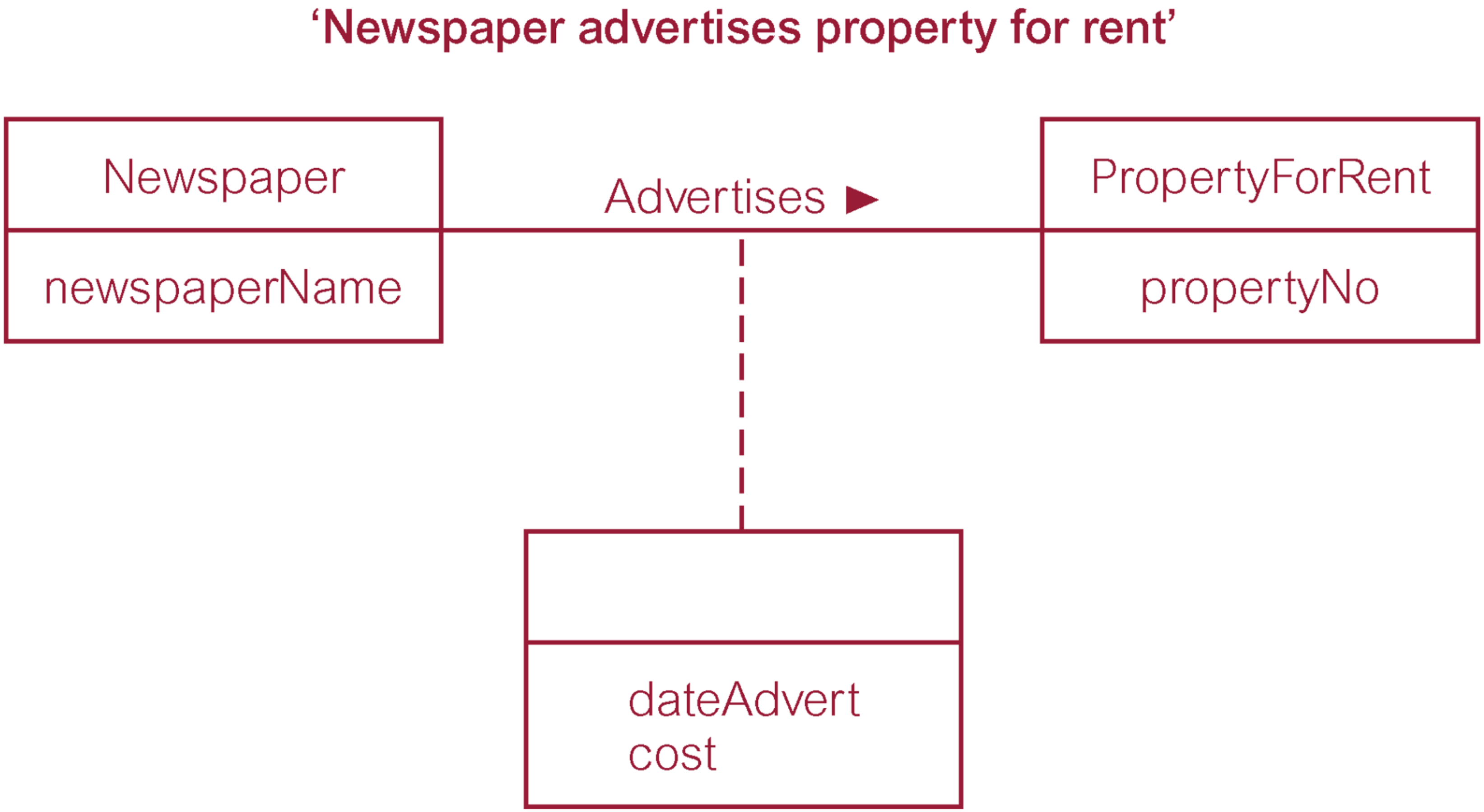
Composite key – multiple attributes compose the key use the tag **{PPK}**

Should also identify the alternate keys with **{AK}**

ER diagram of Staff and Branch entities and their attributes

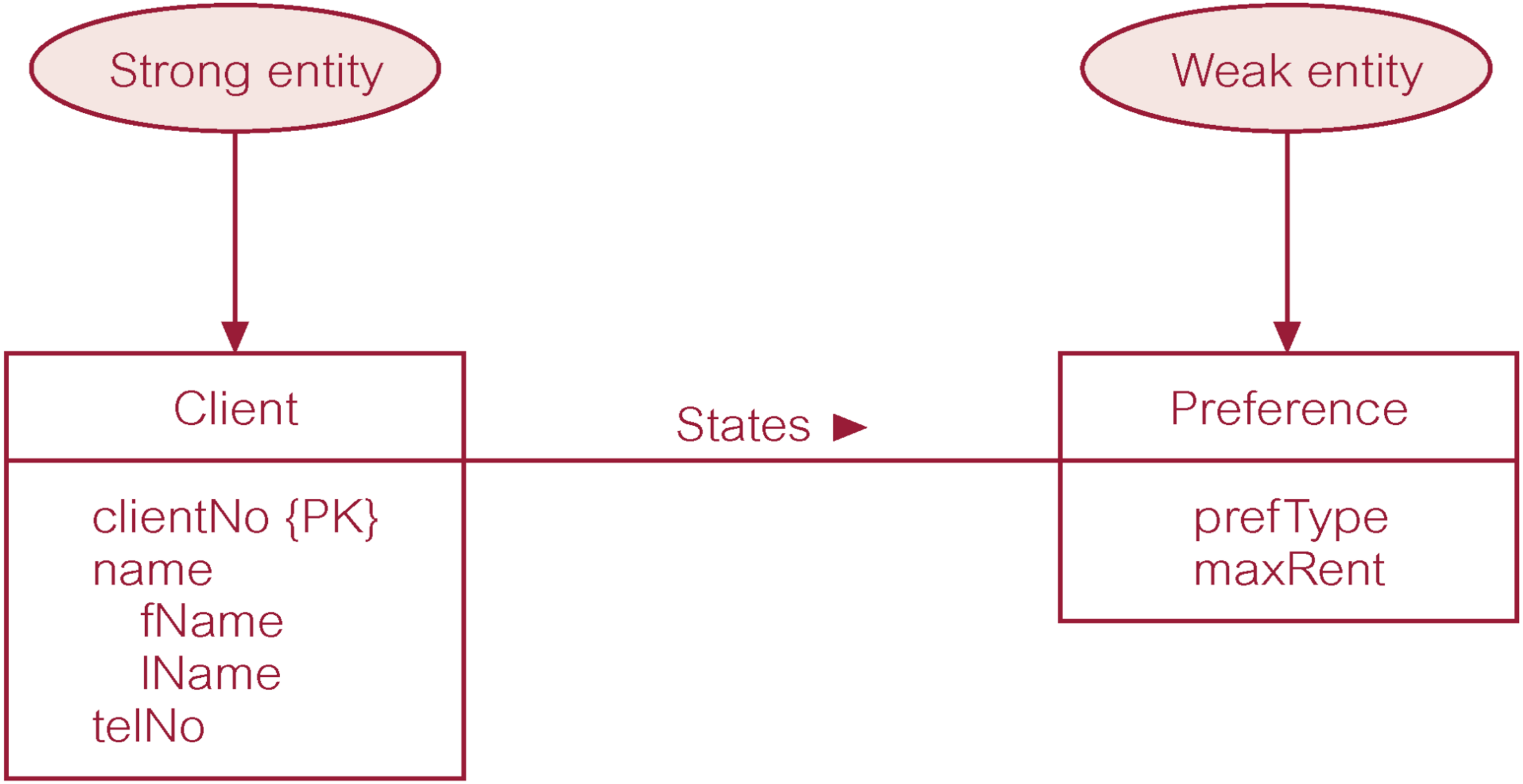


Relationship called *Advertises* with attributes



Strong entity type Client, weak entity type Preference

Can you spot what differentiates a strong from a weak entity?



Modeling exercise

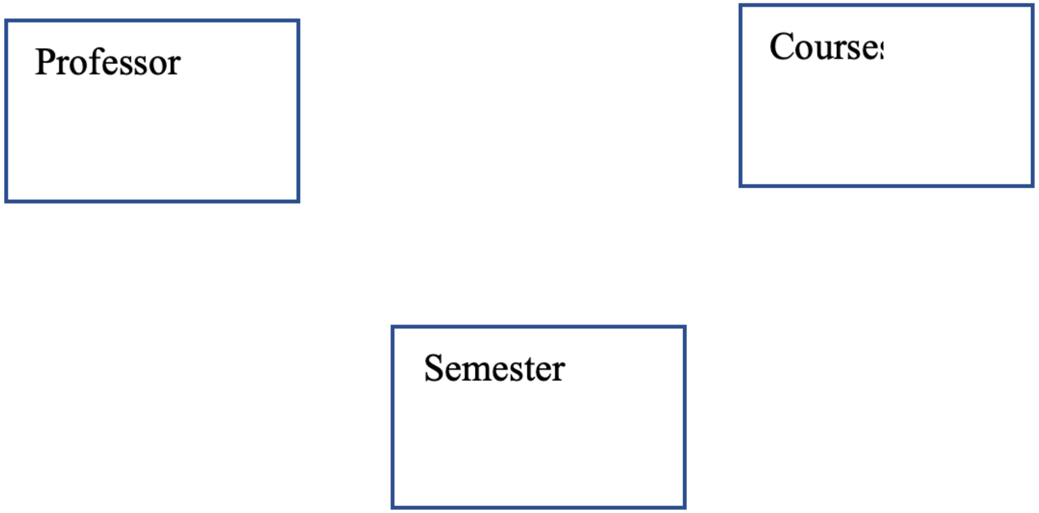
A university database contains information about professors and courses. Professors are identified by social security number, or SSN and courses are identified by course_id. Professors teach one to many courses; a course is taught by one professor. Each of the following scenarios concerns the Teaches relationship between courses and professors. For each of the following scenarios, draw a UML diagram that describes it. Please assume that no other constraints hold.

- Professors can teach the same course in several semesters, and each offering must be recorded (saved in DB).
- Professors can teach the same course in several semesters, and only the most recent such course offering needs to be recorded (saved in DB).

Exercise 1: step 1

Professors can teach the same **course** in several **semester**s, and each offering must be recorded (saved in DB).

Identify the entities.



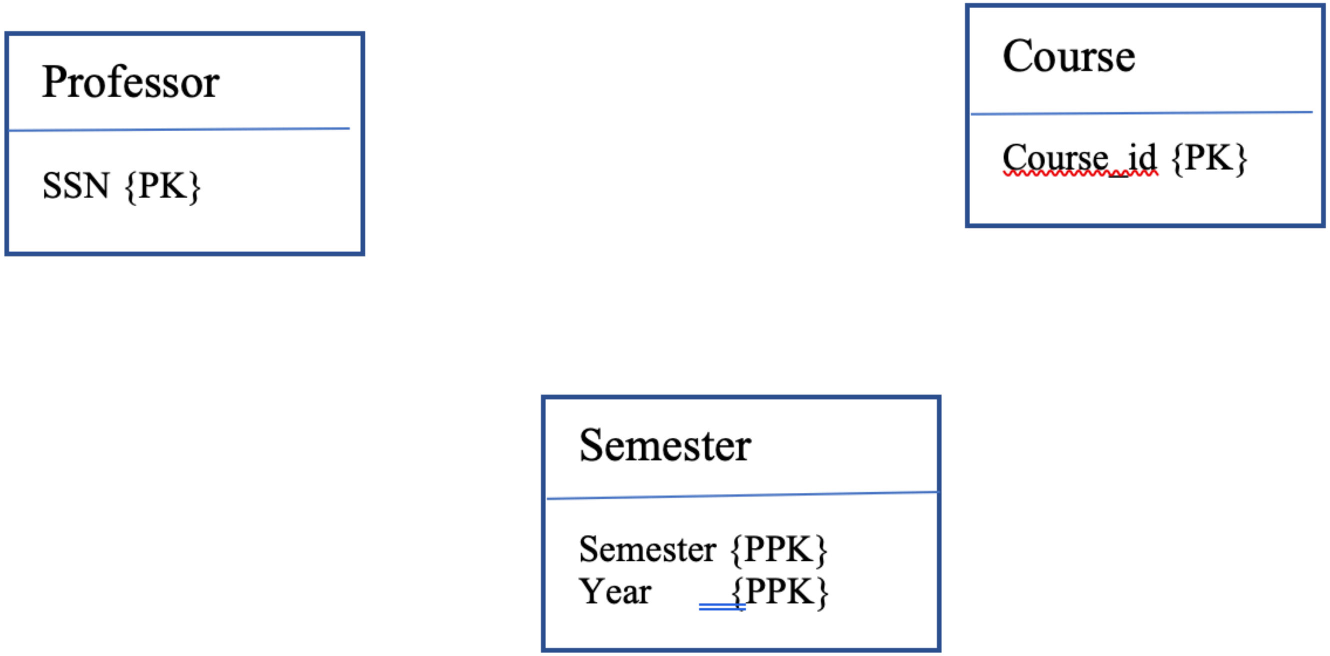
Professor

Courses

Semester

Exercise 1: step 2

Identify attributes and primary keys



Professor

SSN {PK}

Course

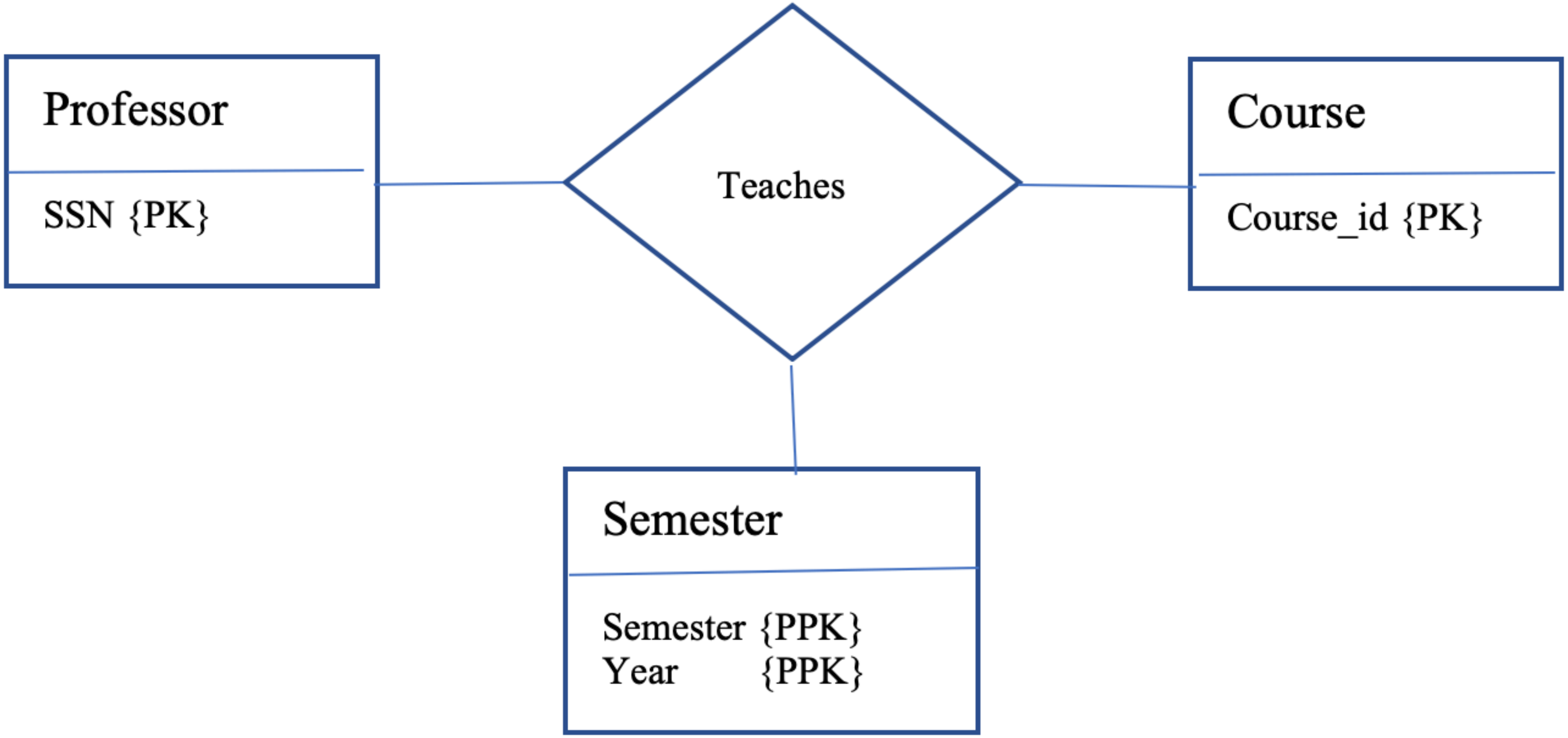
Course_id {PK}

Semester

Semester {PPK}
Year {PPK}

Example 1: Step 3

Professors can teach the same **course** in several **semesters**, and each offering must be recorded (saved in DB).



Exercise 2: step 1

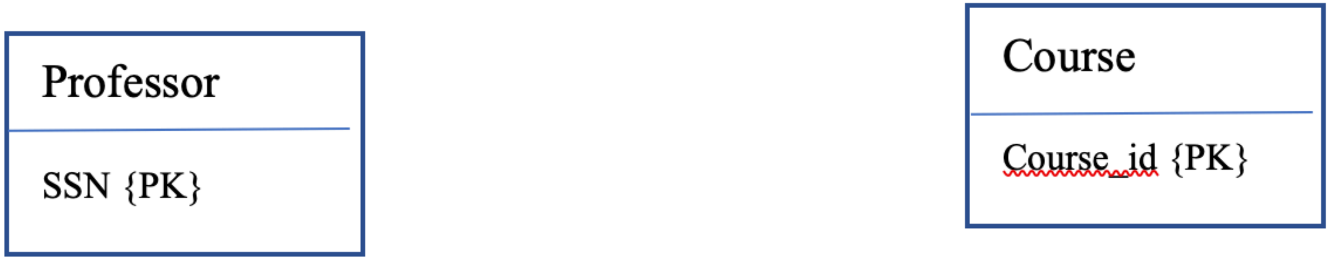
- Professors can teach the same course in several semesters, and only the most recent such course offering needs to be recorded (saved in DB).

  Identify the entities.

Professor
Courses

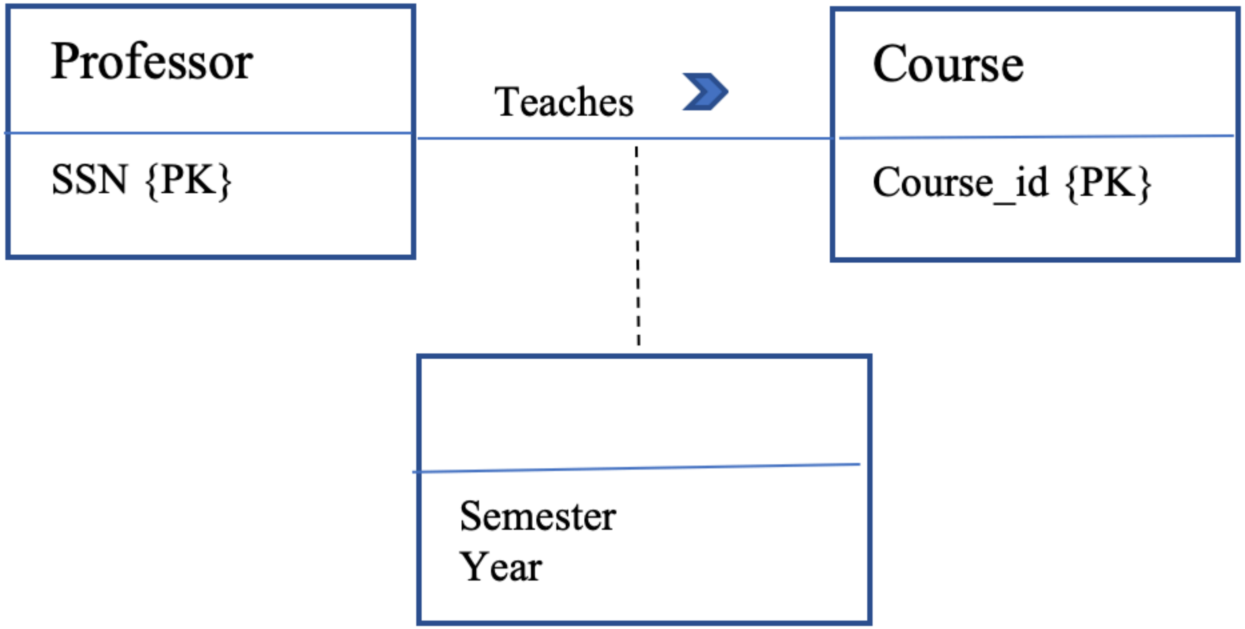
Exercise 2: step 2

Identify attributes and primary keys



Example 2: Step 3

- Professors can teach the same course in several semesters, and only the most recent such course offering needs to be recorded (saved in DB).



Summary

Conceptual modeling identifies the important entities, relationships and attributes for data

There are instances and types (classes) of relationships and entities

Primary keys allow us to distinguish the different entity instances in an entity type

Semantic net models at the object level as opposed to the type or class level