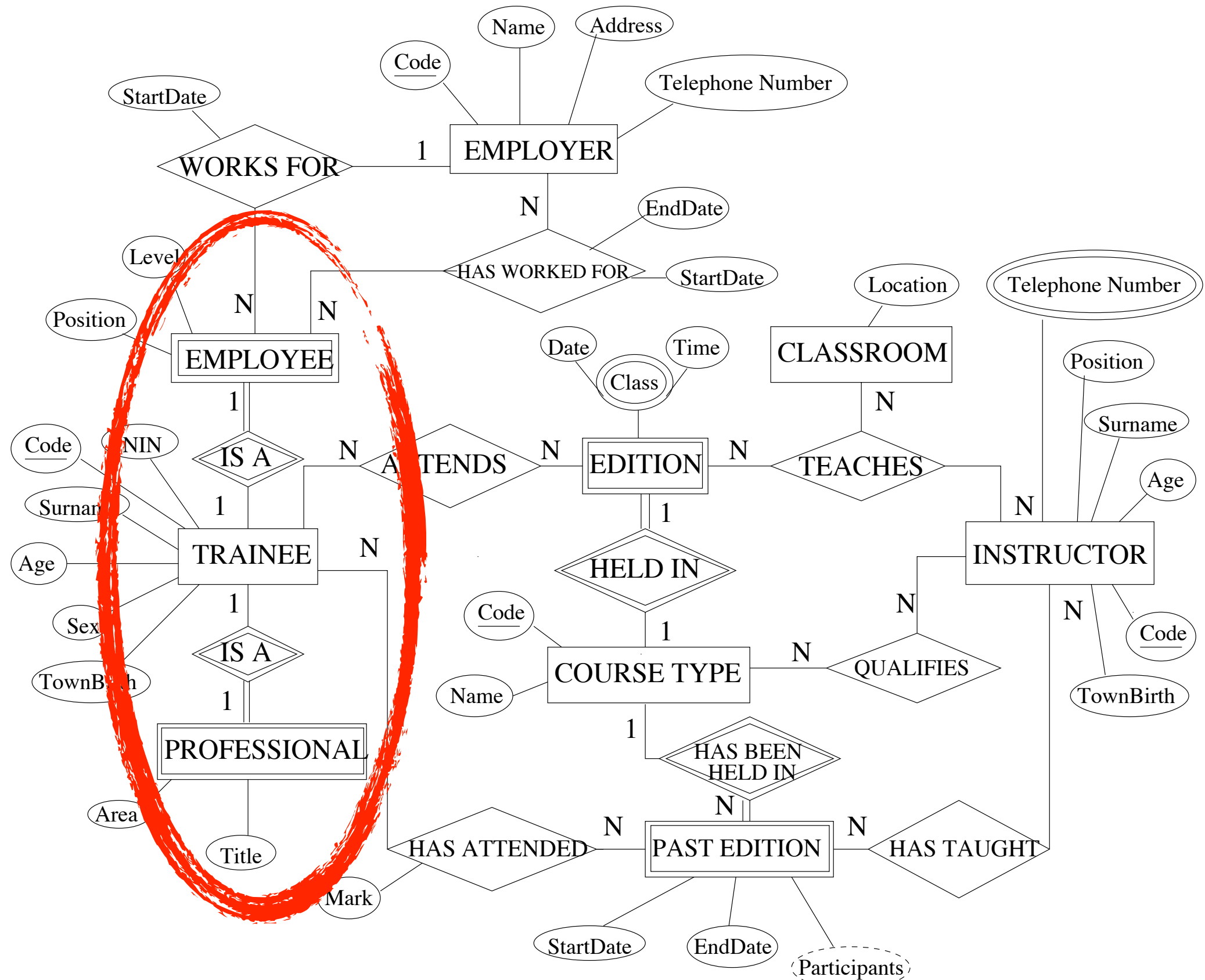


Enhanced ER model

Reading: Elmasri & Navathe Chapter “The Enhanced
Entity-Relationship (EER) Model”

(Chapter 4 in the 7th and 5th edition, Chapter 8 in the 6th edition).



Enhanced ER = EER

- EER model = ER Model + hierarchical relationships
 - entities that are member of one entity type (the *superclass*) may be grouped into meaningful subsets (the *subclasses*)
 - often referred to as an “IS-A” relationship
 - e.g. the two subclasses of TRAINEE in the training courses company (EMPLOYEE and PROFESSIONAL)
- *attribute inheritance* permits economy of representation
 - entity that is a member of a subclass inherits all of the attributes of the entity as a member of the superclass
 - a PROFESSIONAL, as a TRAINEE, has a code, name, etc.
 - also inherits all relationship types the superclass participates in
 - a PROFESSIONAL, as a TRAINEE, attends editions of courses

Specialisation & Generalisation

- Specialisation

- defining a set of subclasses of an entity type on the basis of some distinguishing characteristic of the entities in the superclass

- two main reasons:

1. certain attributes may apply to some but not all members of a entity type:

- e.g. we keep Title and Area for professional trainees only: a subclass is defined to group the entities to which these attributes apply

2. some relationship types may be participating in only by some members of an entity type:

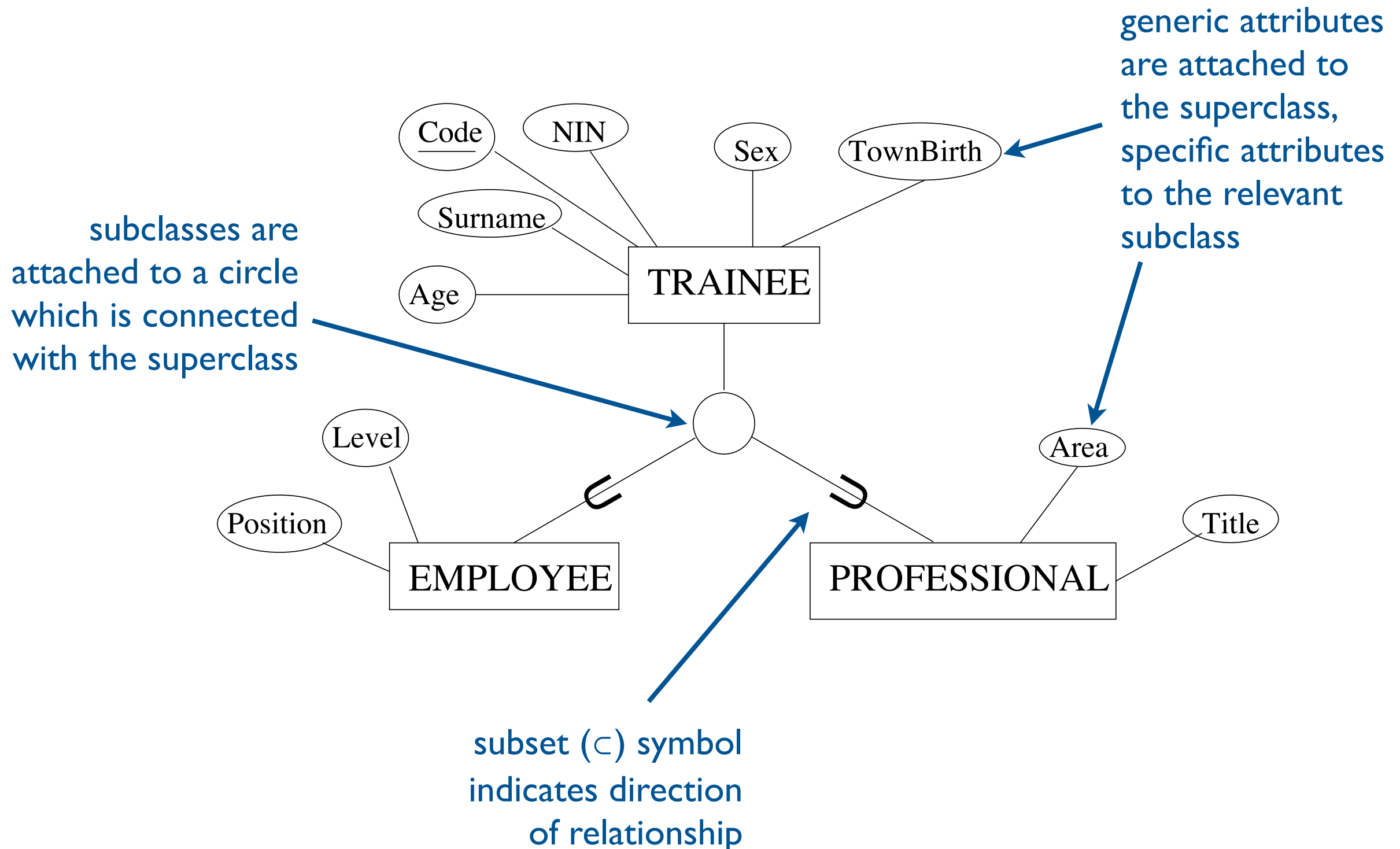
- e.g. employee trainees are the only trainees who works for an employer: a subclass is defined to group the entities participating to this relationship

Specialisation & Generalisation

- **Generalisation**

- Process of defining a generalised entity type from the given entity types
 - inverse of specialisation realise two entity types have lots of common features
- suppress differences between entity types, identify the common features and generalise them into a superclass
 - e.g. generalise CAR and TRUCK into a VEHICLE superclass if a common behaviour is identified
- while some methodologies distinguish between specialisation and generalisation, we will use the same notation
 - they represent two abstraction processes, but the final result is equivalent

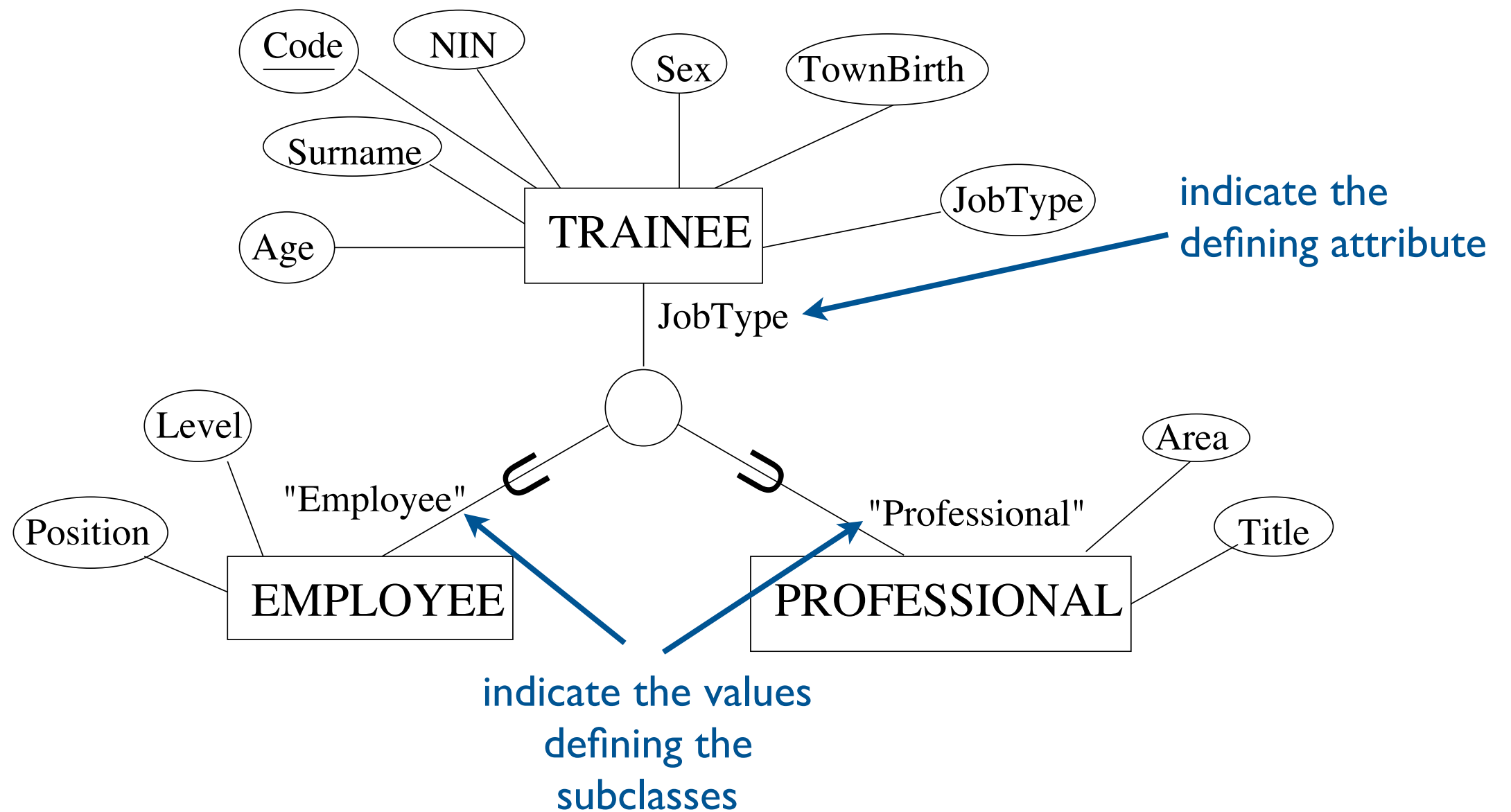
EER notation



Attribute-defined subclasses

- can sometimes determine exactly the entities that will become members of each subclasses
 - e.g. for TRAINEE we may specify the condition of membership to be on the basis of the content of an attribute JobType
- this attribute is the *defining attribute* of the specialisation
- NOTE: all subclasses in a attribute-defined specialisation must have the membership condition on the same attribute

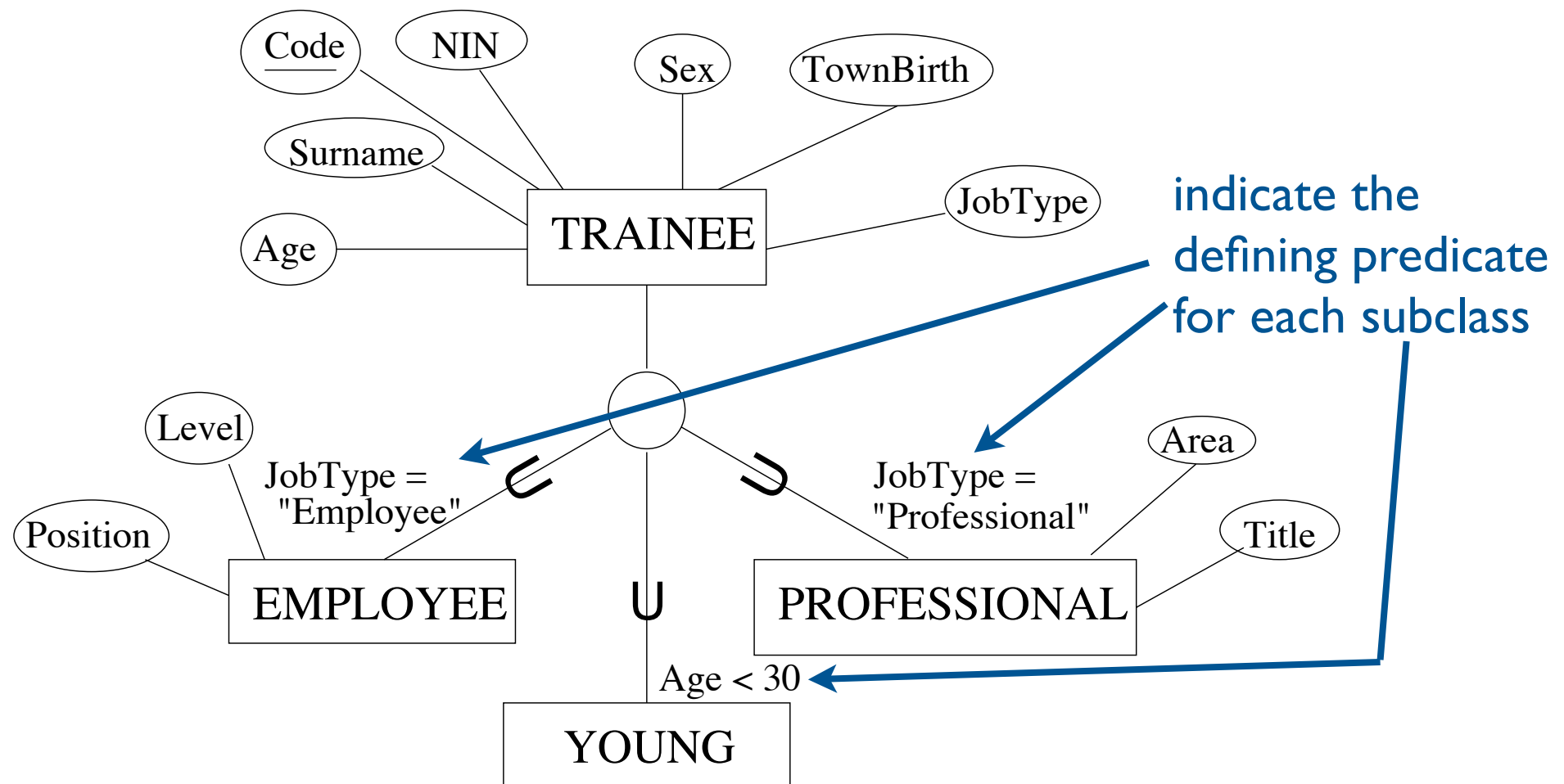
Defining attribute notation



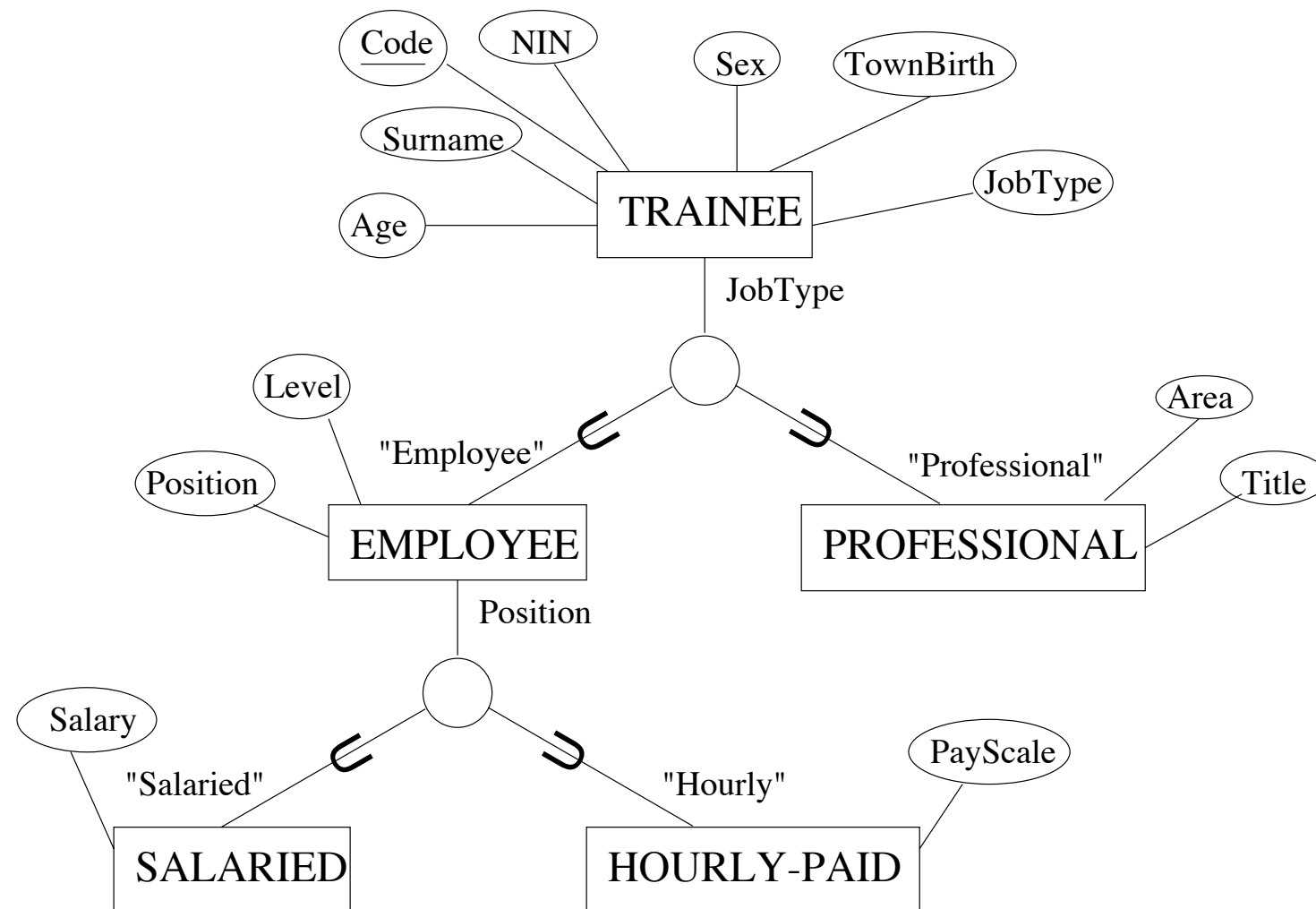
Predicate-defined subclasses

- sometimes the membership condition depends on more than one factor:
 - e.g. for TRAINEE:
 - for EMPLOYEE subclass we may specify the condition of membership to be JobType="Employee"
 - similarly, PROFESSIONALS are those for whom JobType="Professional"
 - but we might have a subclass YOUNG representing all trainees with Age less than 30 (no matter the job): Age < 30
- we are therefore establishing membership on the basis of the truth value of a *defining predicate*

Defining predicate notation



- we can have further specialisations of the same entity type using different distinguishing characteristics e.g. EMPLOYEE can be specialised on the basis of Position into SALARIED and HOURLY-PAID

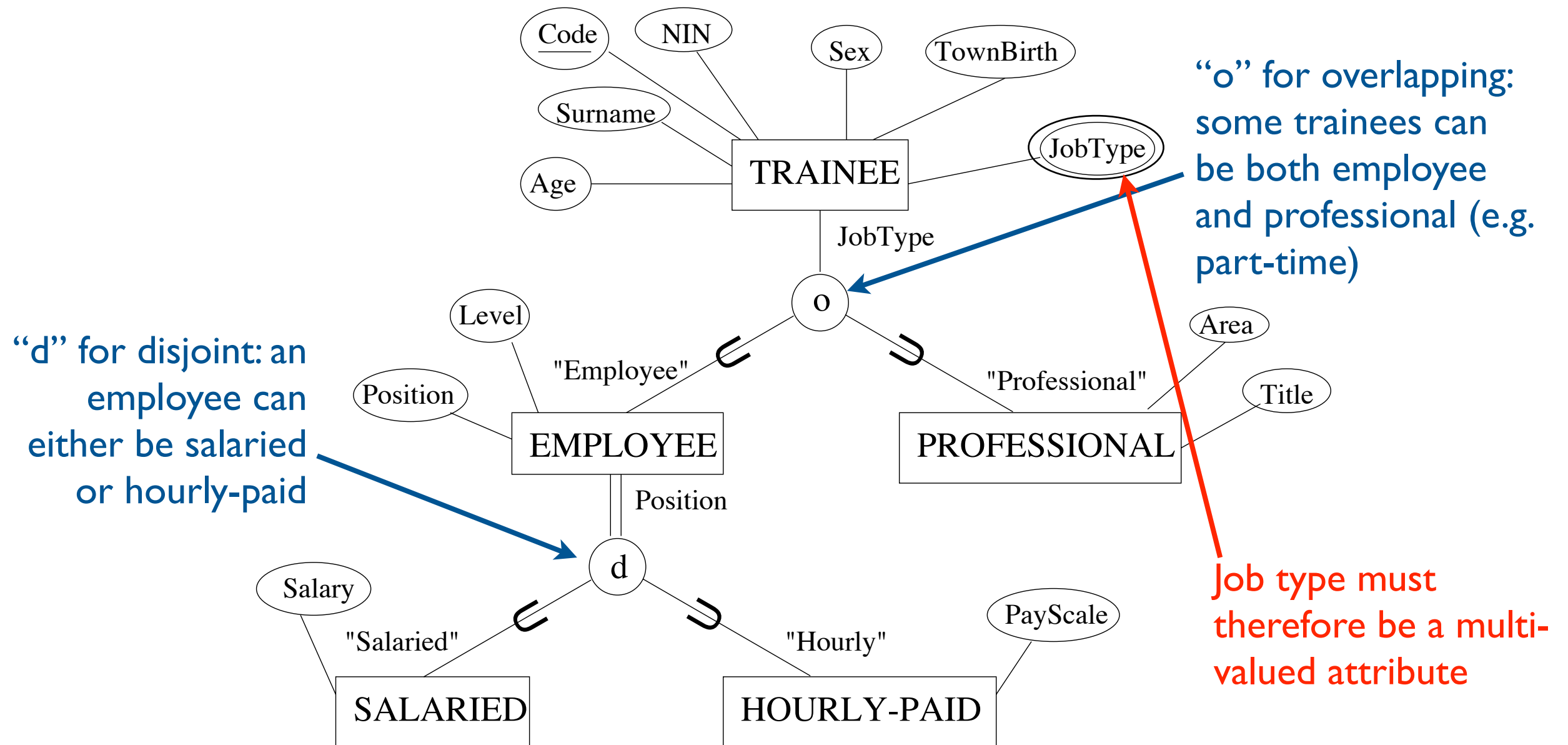


- Note that **SALARIED** and **HOURLY-PAID** are both **EMPLOYEE** and **TRAINEE**

Disjointness Constraint

- specifies that subclasses of a specialisation are **disjoint**
 - an entity can be a member of at most one of the subclasses of the specialisation
 - attribute defined specialisation implies disjoint subclasses if the defining attribute is single-valued
- subclasses that are not disjoint may **overlap**
 - some entities may be member of more than one subclass of the specialisation

disjoint/overlap notation



Completeness constraint

- Same as ordinary relationships:

1. *Total Specialisation*: every entity in a superclass must be a member of some subclass in some specialisation

- e.g. every EMPLOYEE must be either HOURLY-PAID or SALARIED

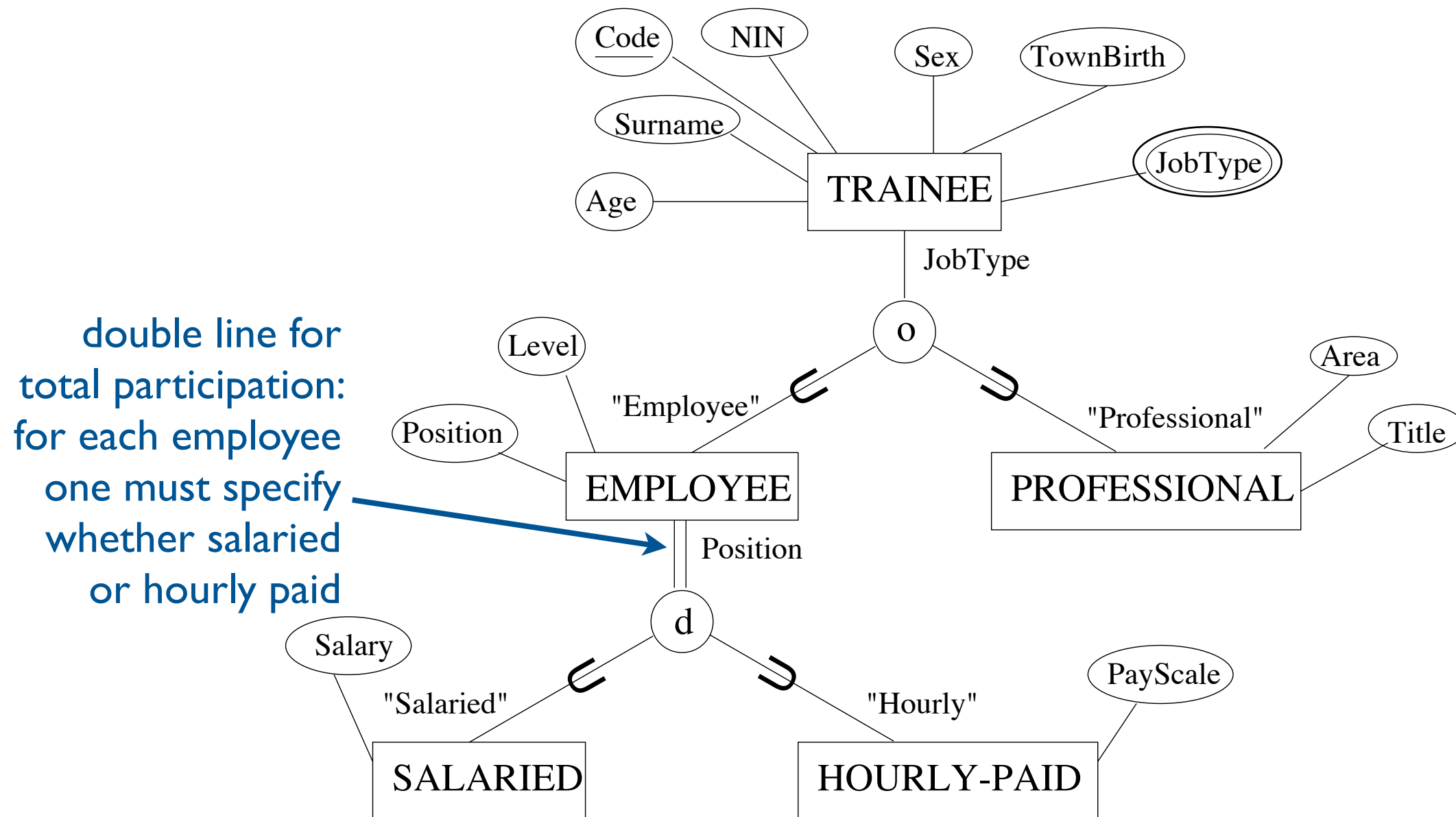
2. *Partial Specialisation*: allows an entity not to belong to any of the subclasses

- e.g. can have TRAINEES who are neither EMPLOYEES nor PROFESSIONALS

- Note: disjointness and completeness are *independent*:

- can have Disjoint+Total, Disjoint+Partial, Overlapping+Total and Overlapping+Partial relationships

total/partial participation



Insertion & Deletion Rules

1. deleting an entity from a super class implies that it is automatically deleted from all of the subclasses it belongs to
2. inserting an entity in a superclass implies that the entity is inserted in all predicate-defined subclasses for which the entity satisfies the defining predicate
3. inserting an entity in a superclass of total specialisation implies that the entity is inserted in at least one of the subclasses of the specialisation
4. inserting an entity in a super class of disjoint, total specialisation implies that the entity is inserted in one and only one of the subclasses of the specialisation

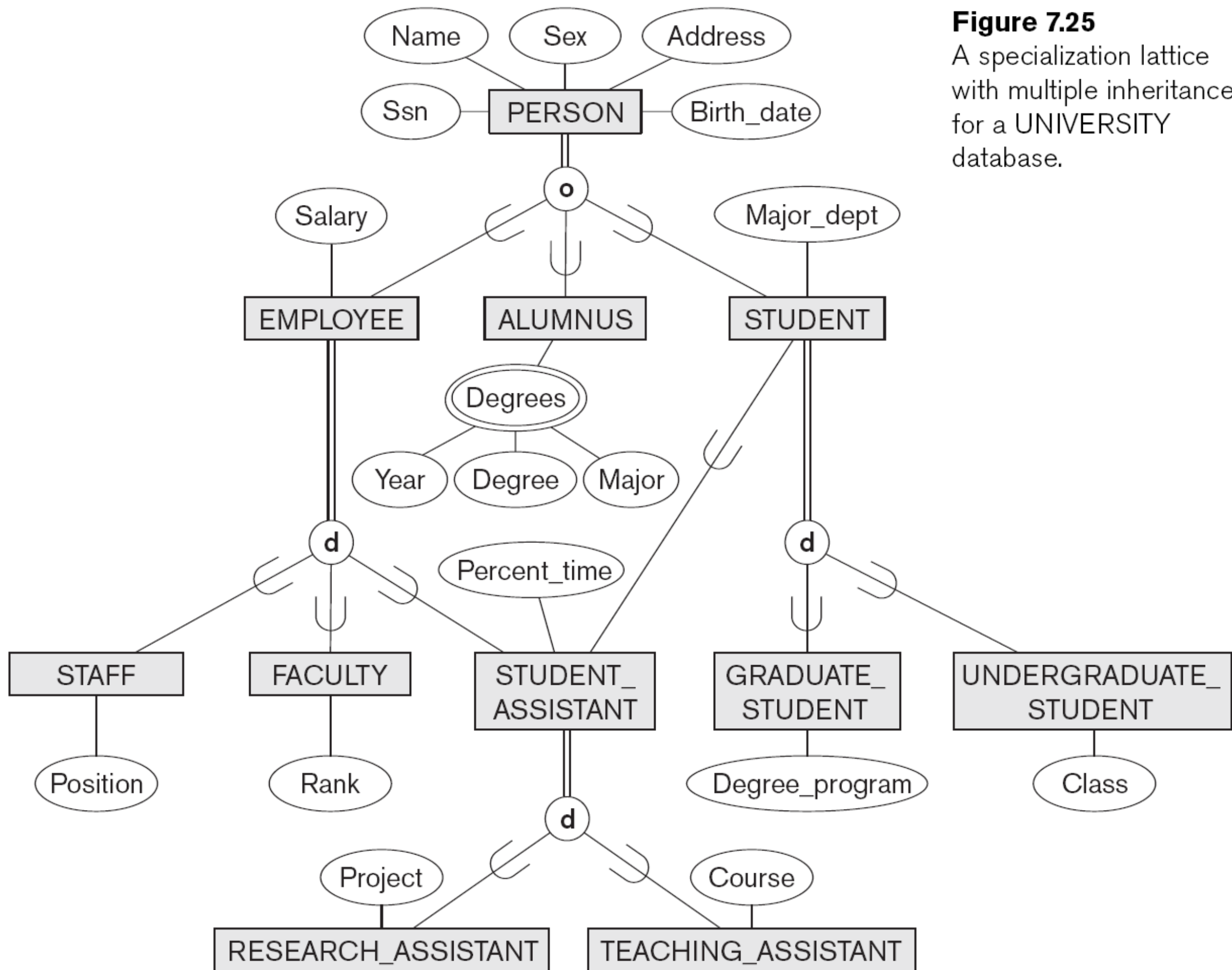


Figure 7.25

A specialization lattice with multiple inheritance for a UNIVERSITY database.