COS221
L05 - Enhanced Entity-Relationship Modelling (Part 1)

(Chapter 8 in Edition 6 and Chapter 4 in Edition 7)

Linda Marshall

2 March 2023

Semantic data modelling in Computer Science

Various semantic data models have been introduced into computer science

- Knowledge representation in AI
- Object modelling in SE
- Enhanced ER for Database systems.
  This includes concepts such as:
  - class/subclass relationships
  - type inheritance
  - specialisation/generalisation and constraints
  - modelling UNION contracts with categories
  - ...

Subtypes (or Subclasses)

In ER-modelling (L03 and L04),

- An entity type represented both a *type of entity* and an *entity set* (or *collection of entities of that type*) that exist in the database. For example, EMPLOYEE describes the type (attributes and relationships) and the current set of employee entities in the COMPANY database.

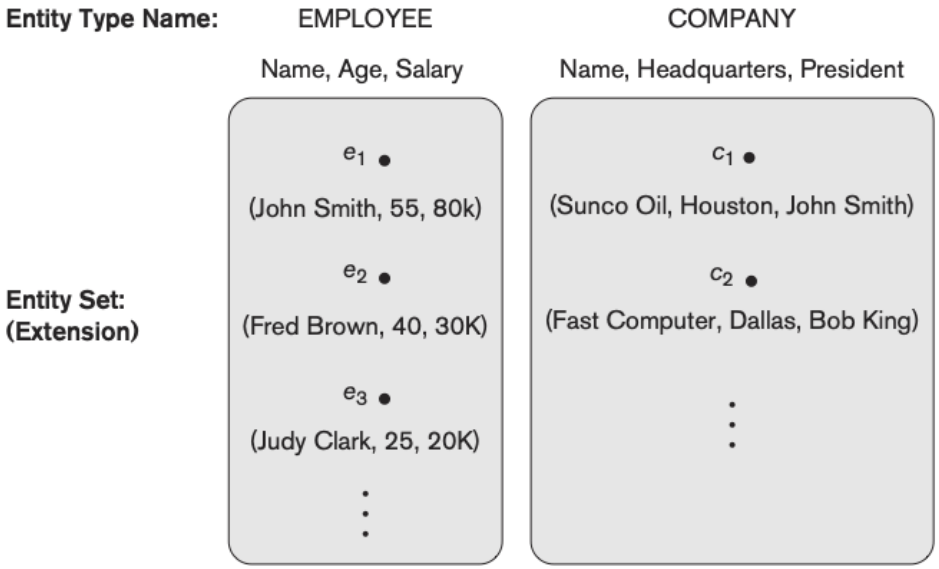


Figure 3.6
Two entity types, EMPLOYEE and COMPANY, and some member entities of each.

- There are relationships between entities.

Subtypes (or Subclasses)

- An entity type may have many *subgroupings* (or *subtypes or subclasses*) that are meaningful. For example, Employees can be differentiated by post-level (secretary, manager, ...), or employment contract (part-time, ...). The set of entities in the subgrouping is a subset of the entities belonging to the EMPLOYEE entity set.
  - A 'parent' type/class of a subtype is called *supertype* or *superclass*
  - A subtype/class has an 'is-a' relationship with it supertype/class.
- Because the entity of the subtype represents the same real-world entity in the superclass, it possesses (*inherits*) all the attributes associated with the superclass. This is referred to as *type inheritance*.

Specialisation and Generalisation

A *specialisation* defines a set of subclasses of an entity type - the superclass of the specialisation. In the figure below there are three (3) specialisations of EMPLOYEE shown.

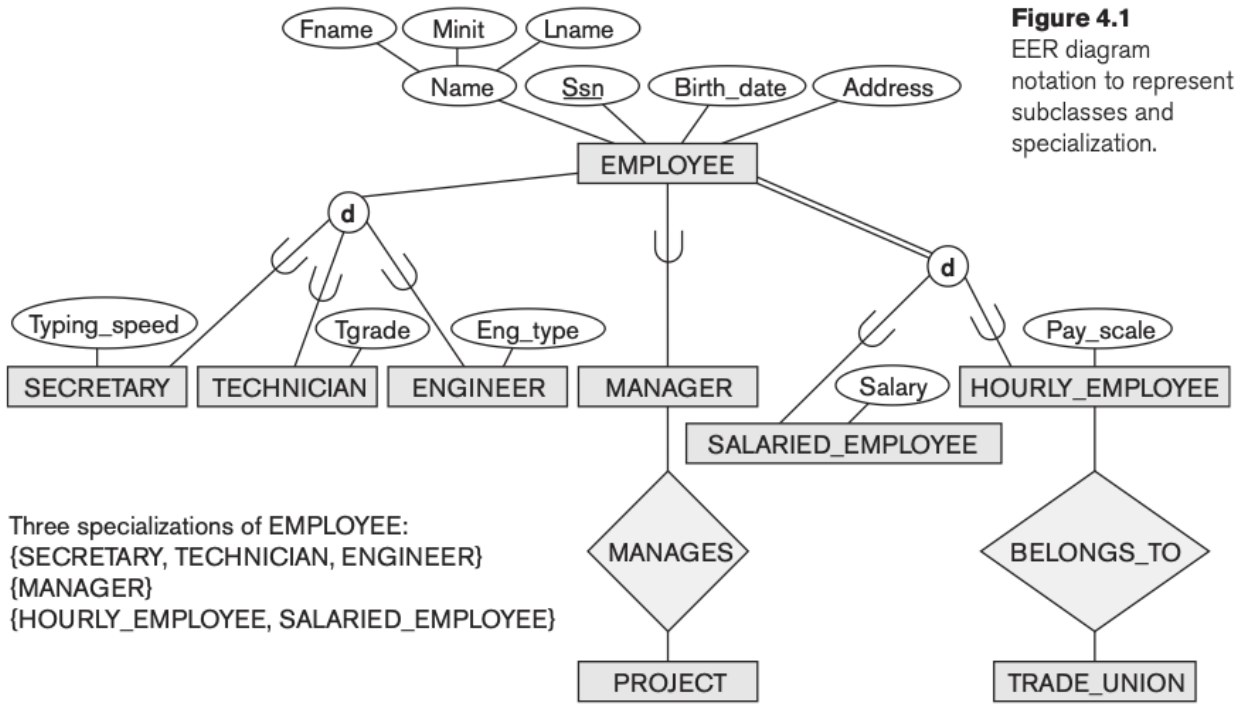


Figure 4.1
EER diagram notation to represent subclasses and specialization.

Specialisation and Generalisation - Notation

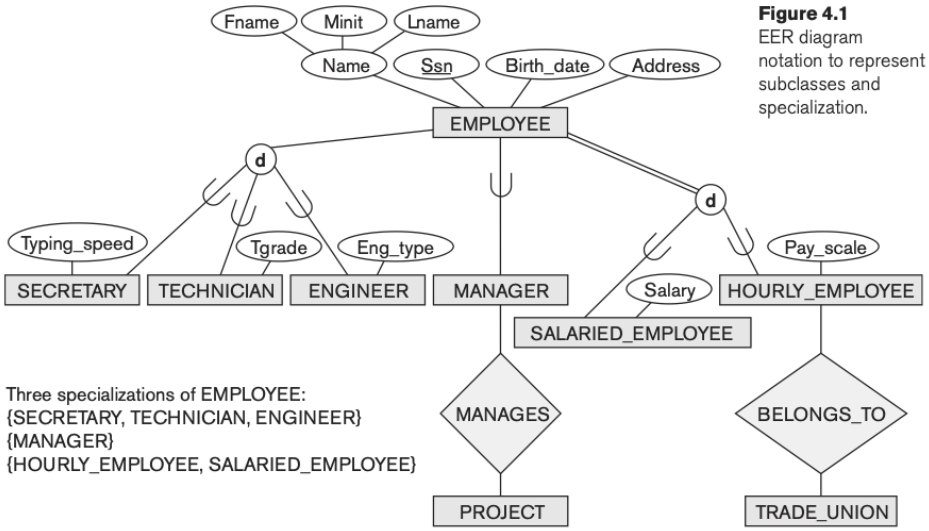


Figure 4.1
EER diagram notation to represent subclasses and specialization.

- Specialisations are linked to the superclass with a line to a circle then another line to the superclass.
- The *proper subset symbol* on the lines shows the direction of the subset/superclass relationship.
- Subsets may include attributes specific to the subclass, referred to as *specific* or *local* attributes.

Specialisation and Generalisation - Notation

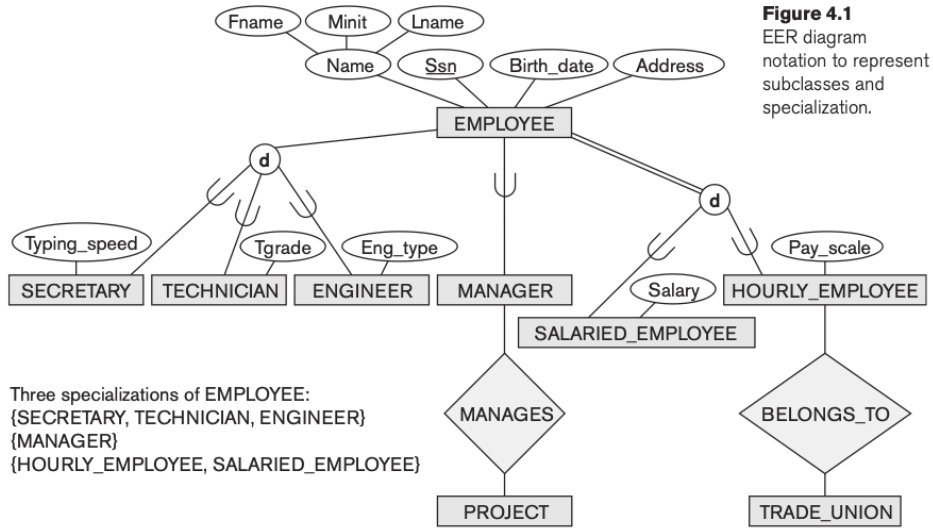


Figure 4.1
EER diagram notation to represent subclasses and specialization.

- A specialisation may comprise of multiple entity types with the same characteristics (SECRETARY,TECHNICIAN and ENGINEER) or a single entity type (MANAGER)
- A specialisation with a single entity type is drawn using only a line. That is, the circle notation is not used.

Specialisation and Generalisation - Notation

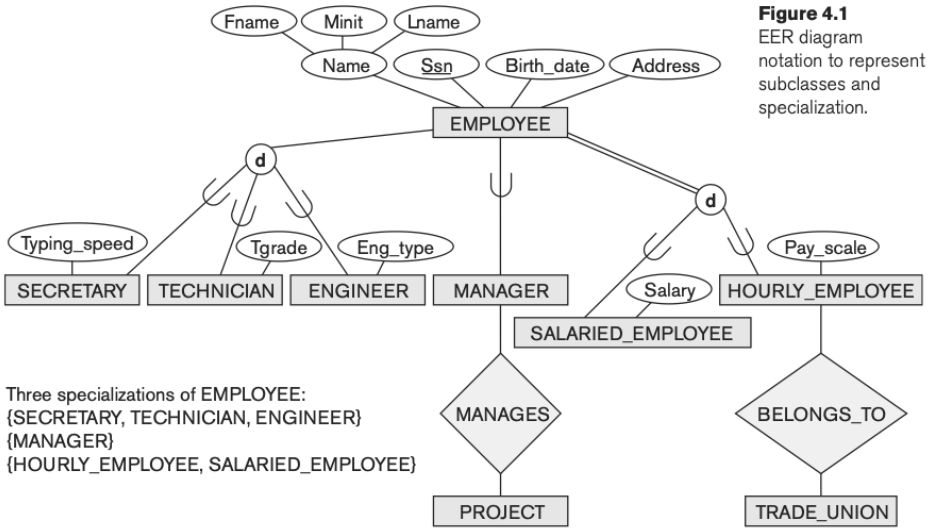


Figure 4.1
EER diagram notation to represent subclasses and specialization.

- Specialisations are used when some relationships may be participated in and others not.
- An HOURLY_EMPLOYEE may belong to a trade union.

Specialisation and Generalisation - Instances of a Specialisation

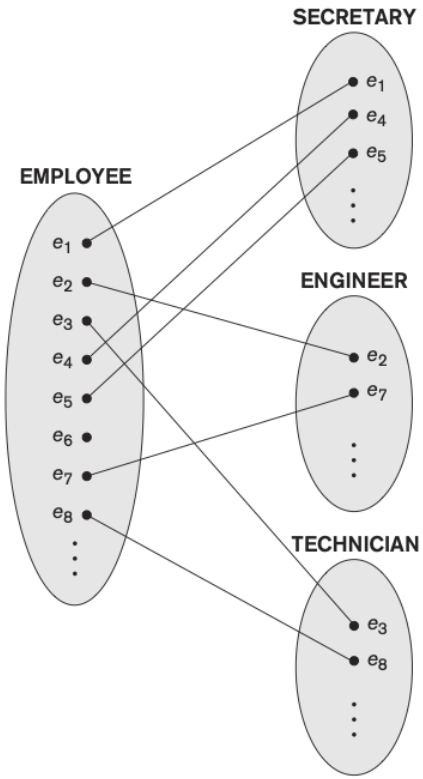


Figure 4.2
Instances of a specialization.

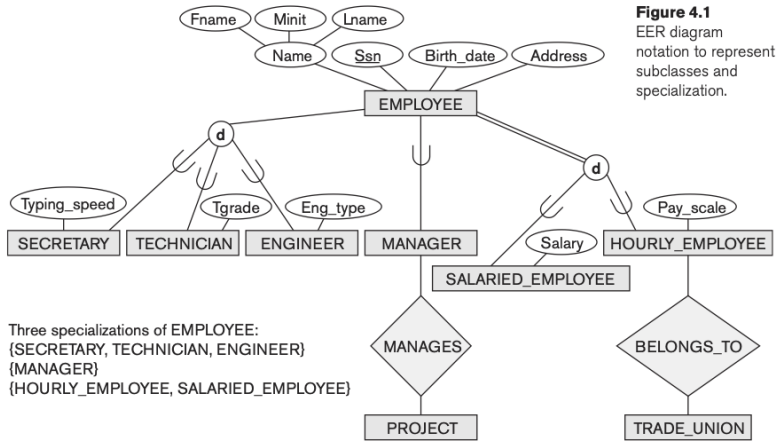
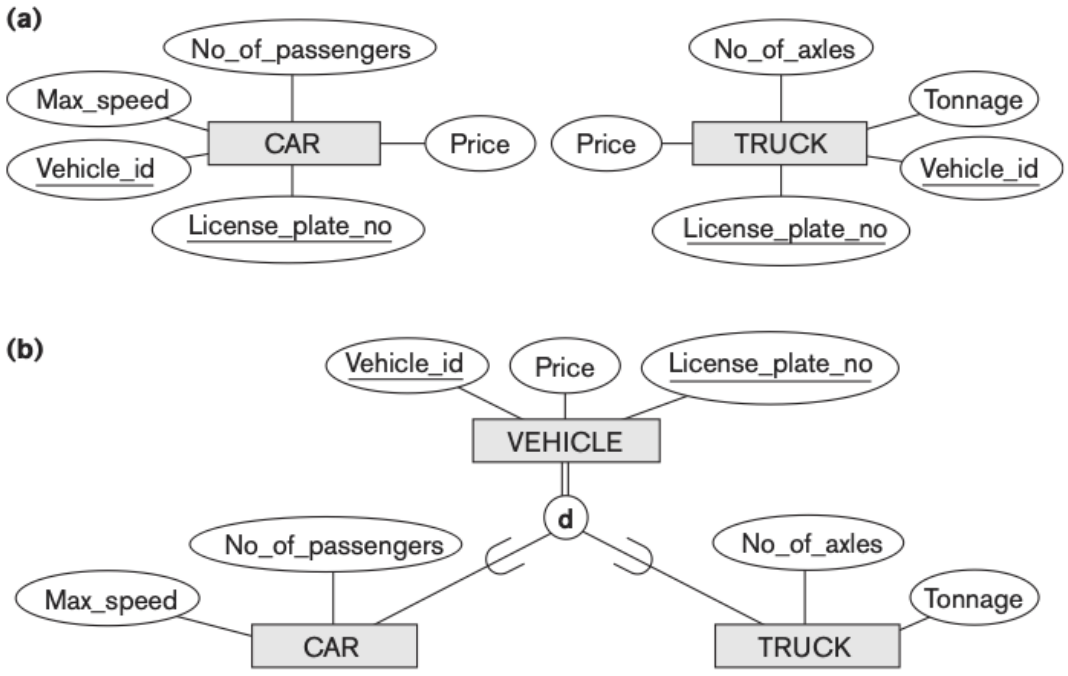


Figure 4.1
EER diagram notation to represent subclasses and specialization.

Specialisation and Generalisation

Figure 4.3
Generalization. (a) Two entity types, CAR and TRUCK.
(b) Generalizing CAR and TRUCK into the superclass VEHICLE.



- The reverse process of specialisation is generalisation.
- Given two or more related entities that share common attributes, these common attributes can be generalised (abstracted out) to form the superclass.

Constraints on Specialisation/Generalisation Hierarchies

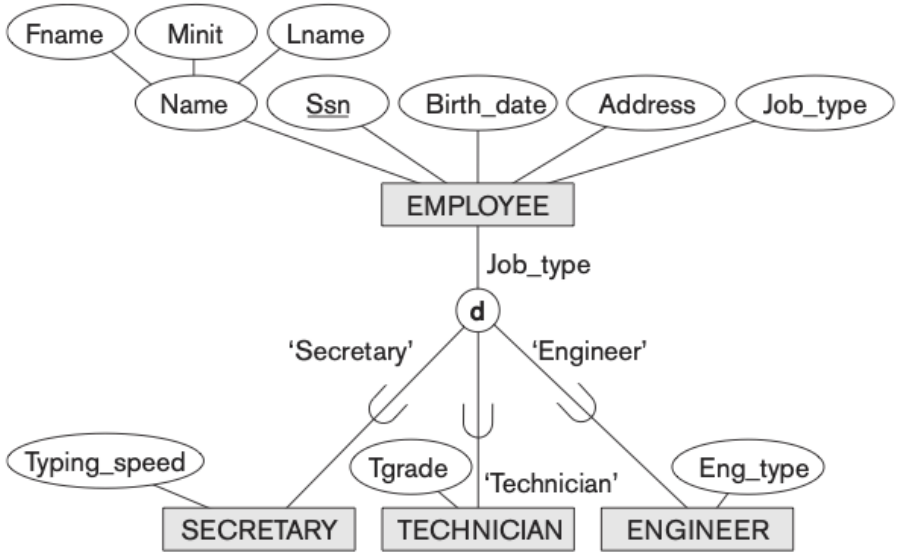


Figure 4.4
EER diagram notation for an attribute-defined specialization on Job_type.

- Placing a constraint on the value of some member of the superclass will allow us to specify exactly which entities will become members of the subclass.
- These subclasses are referred to as *predicate-defined* (or *condition-defined* subclasses.
- `Job_type = 'Secretary'`, is called the *defining predicate* and `Job_type` in EMPLOYEE, the *defining attribute*.
- The specialisation if termed an *attribute-defined specialisation*.

Constraints on Specialisation/Generalisation Hierarchies

- When no condition is defined to determine membership of subclasses, the subclass is called *user-defined*.
- Deciding membership of the subclass in this case is a manual process and determined by the user.
- The process cannot be automated.

Constraints on Specialisation/Generalisation Hierarchies

- Other constraints that may apply to a specialisation
  - Disjointness (or disjointedness)
  - Completeness (or totalness)
- Disjointness and Completeness are independent and therefore the following combinations are possible:
  - Disjoint and Total
  - Disjoint and Partial
  - Complete and Total
  - Complete and Partial

Constraints on Specialisation/Generalisation Hierarchies - Disjointness

- An entity of the specialisation may be a member of at most one subclass of the specialisation.
- An *attribute-defined* specialisation implies disjointedness.
- Disjointedness is specified using a **d** in the circle.
- Sets of entities of subclasses that are not disjoint are said to be *overlapping*. An **o** in the circle is the designation for this, the default, case.

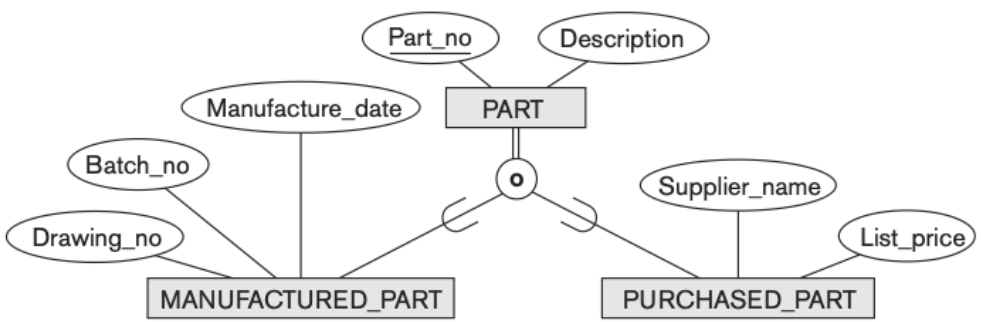


Figure 4.5
EER diagram notation for an overlapping (nondisjoint) specialization.

Constraints on Specialisation/Generalisation Hierarchies - Completeness

The completeness constraint may be *partial* or *total*.

- A *partial specialisation* means that the entity need not be a member of one of its subclasses.
  - This is shown on the ER-diagram using a single line.
  - For example, an EMPLOYEE may be a SECRETARY, ENGINEER, TECHNICIAN or MANAGER.
- A *total specialisation* means that every entity in the superclass **must** be a member of at least one subclass.
  - This is shown using a double line on the ER-diagram.
  - For example, an EMPLOYEE must be either a SALARIED_EMPLOYEE or an HOURLY_EMPLOYEE.

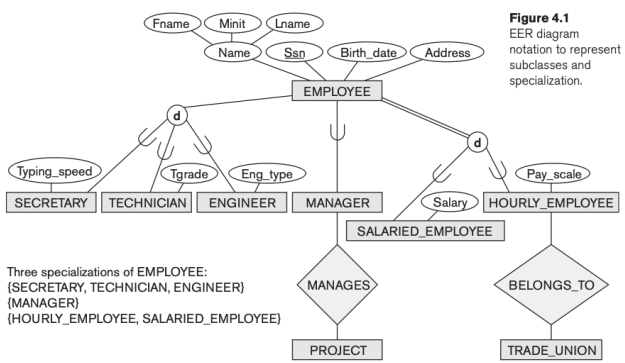


Figure 4.1
EER diagram notation to represent subclasses and specialization.