


COSC210 Database Management Systems
Lecture 5 - The Enhanced Entity-Relationship Model
 Dr. Mitchell Welch

1 / 34

Reading

- Chapter 4 from *Fundamentals of Database Systems* by Elmasri and Navathe

Summary

- ER Model vs EER Model
- EER Model - Object Oriented Approach
- Specialisation
- Generalisation
- Constraints of Specialisation and Generalisation
- Hierarchies and Lattices
- Modeling Union Types
- UNIVERSITY Example
- Design Choices

ER Model vs EER Model

- The EER Model includes all of the modelling concepts present within the ER Model.
- The EER Model introduces the concept of *object-oriented design* to ER modelling
- EER Models introduce the term *Object* which is used interchangeably with *Entity*.

ER Model vs EER Model

- EER Models include the concepts of:
 - Superclasses
 - Subclasses
 - Generalisation
 - Specialisation
 - Category/Union-type
- These concepts are analogous to the features presented within a typical object-oriented programming language.

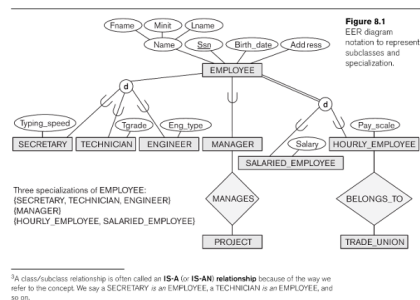
5 / 34

EER Model - Object Oriented Approach

- A *Subclass* or *Subtype* of an entity type allows for the specification of sub-groupings within the entity type.
- For example our EMPLOYEE entity could be distinguished further into SECRETARY, ENGINEER and MANAGER entities.
- In this example, the EMPLOYEE entity is the superclass (or supertype) and each of the sub-groupings constitutes a subclass.
- This represented within the EER model as a *superclass/subclass* relationship.
- This approach reflects the implementation within the relational database, using a foreign key relationship between the sub and super classes

6 / 34

EER Model - Object Oriented Approach



7 / 34

EER Model - Object Oriented Approach

- An entity cannot exist in a database as merely a sub-type object. (i.e. a Member of a subclass must be a member of some superclass)
- An entity can be a member of any number of subclasses.
- For example, from the model in the previous slide, it is possible to have an entity that belongs to the SALARIED_EMPLOYEE and ENGINEER subtypes of EMPLOYEE.
- It is possible for a superclass object to exist without being a member of a subclass.

8 / 34

EER Model - Object Oriented Approach

- Another important Object-Oriented design feature include in the EER model is **type inheritance**
- Recall that an entity type is defined by the attribute set that it possesses and the relationship-types that it participates in.
- In an EER model, subclass objects *Inherit* all the attributes of their superclasses **in addition** to attribute set unique to the subclass type.

Specialisation

- **Specialisation** allows for the definition of sets of subclasses.
- The set of subclasses is defined on the basis of some common characteristic.
- The the EMPLOYEE example there are two specialisations:
 - One the basis of *job type*
 - The second is on the basis of *method of pay*
- Specialisation is represented using a circle with lines attached to the subtypes and a single line back to the supertype.
- The *subset* symbol on each of the lines to the subtypes represents the direction of the relationship in this notation.
- Attributes that apply only to the subtypes are attached directly to the subtype entities and are referred to as **specific attributes**

Specialisation

- Specialisation allows for entities to be grouped according to common attributes.
- Entities may share the majority of their attributes - with only a small number of specific attributes on the individual subclasses
- Entities within the subclasses may participate in different relationships to the superclass and other subclasses.
- Overall, the concept of specialisation provides an efficient way of representing related but distinct object types.

Generalisation

- Generalisation is effectively the reverse process of specialisation.
- In this process, we identify common features amongst entities and consolidated them into a superclass
- The original entities are organised as subclasses and specialisations, with their remaining unique attributes sets included as specific attributes.
- For example, the two entities CAR and TRUCK can be generalised into the entity VEHICLE.
- The common attributes are placed onto the superclass (VEHICLE)

Generalisation

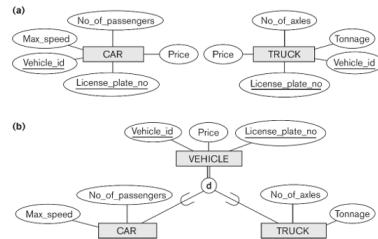


Figure 8.3
Generalization. (a) Two entity types, CAR and TRUCK. (b)
Generalizing CAR and TRUCK into the superclass VEHICLE.

Constraints on Specialisation and Generalisation

- It is possible to represent *condition-defined* subclasses that are dependent on the value of a superclass attribute.
- For example, if we add the attribute *job_type* to our EMPLOYEE supertype, we can make membership within the subclasses conditional upon its value.
- If all subclasses within a specialisation have their membership conditional on the same attribute, the specialisation is said to be **attribute-defined**
- When there is no condition on the membership, the specialisation is **user-defined**

Constraints on Specialisation and Generalisation

- Attribute-defined specialisation is a *constraint* on the model.

Constraints on Specialisation and Generalisation

- The disjoint constraint specifies that subclasses of a specialisation must be disjoint.
- This means that an entity can be a member of at most one subclass.
- Attribute-defined specialisations already imply disjointedness.
- This constraint is denoted with a 'd' in the specialisation grouping circle.
- Specialisation that do not have the disjoint constraint can have entities that belong to multiple subclasses.
- This is denoted with an 'o' in the specialisation grouping circle.

Figure 8.4
EER diagram notation



Constraints on Specialisation and Generalisation

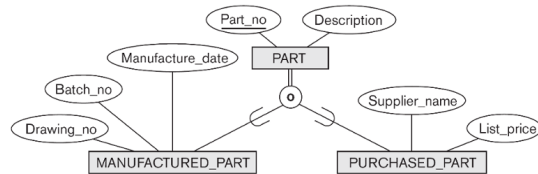


Figure 8.5
EER diagram notation
for an overlapping
(nondisjoint)
specialization.

⁷The notation of using single or double lines is similar to that for partial or total participation of an entity type in a relationship type, as described in Chapter 7.

Constraints on Specialisation and Generalisation

- The **total specialisation constraint** specifies that each member in the superclass must be a member of at least one subclass in the specialisation.
- This is denoted by the double line connecting the superclass to the specialisation.
- A single line indicates a **partial specialisation** which allows an entity to not belong to any of the subclasses.

Constraints on Specialisation and Generalisation

- The combination of these constraints allows for four possible combinations:
 - Disjoint, total
 - Disjoint, partial
 - Overlapping, total
 - Overlapping partial
- The combinations used will be determined by the application
- Generally, superclasses that are identified through the generalisation process will be total because the superclass is derived from the subclasses.

Constraints on Specialisation and Generalisation

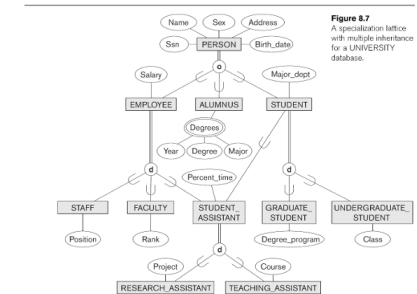
- Within a generalisation/specialisation relationship, there are rules that apply for insertion and deletion:
 - Deletion of the superclass implies that it is deleted from all subclasses
 - Inserting an entity into a superclass means that it is inserted into subclasses as set by the *defining-predicate* and attribute values
 - Inserting into a superclass with total specialisation implies that it is inserted into at least one subclass of the specialisation.

Hierarchies and Lattices

- Specialisation/generalisation relationships can be used to form *hierarchies* and *lattice* structures.
 - A hierarchy structure is constrained by allowing the subclass to participate in only one relationship
 - A Lattice allows subclasses to belong to more than one relationship.
- In both cases a subclass inherits the attributes not only from its direct superclass, but from all predecessors right back through to the *root* node of the specialisation.
- a *leaf node* is a class that has no subclasses.

21 / 34

Hierarchies and Lattices



22 / 34

Hierarchies and Lattices

- In a lattice structure, it is possible for a subclass to inherit the same attributes twice via different paths in the lattice.
- For example the STUDENT_ASSISTANT entity will inherit the PERSON attributes twice.
- In this situation, the attributes are only included on the subclass once.

23 / 34

Hierarchies and Lattices

- Lattices and Hierarchies can be generated using a top-down or bottom-up refinement process.
 - Using a top-down approach, we start with the superclasses and specialise down to the subclasses
 - Using the bottom-up approach, we start with the individual entity-types as subclasses and generalise according to the common attributes to create the superclasses.
- These processes (when applied to the same application) can create identical arrangements of superclasses and subclasses.

24 / 34

Modeling Union Types

- Up until this point, all relationships we have looked at have had a single superclass.
- Sometimes it may be necessary to use multiple superclasses
- In this situation, the subclass will represent a collection of objects that is a subset of the union of distinct entity types.
- This is denoted using the 'U' operator on the link between the relationship and the subclass.

Modeling Union Types

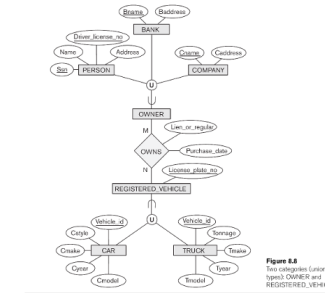
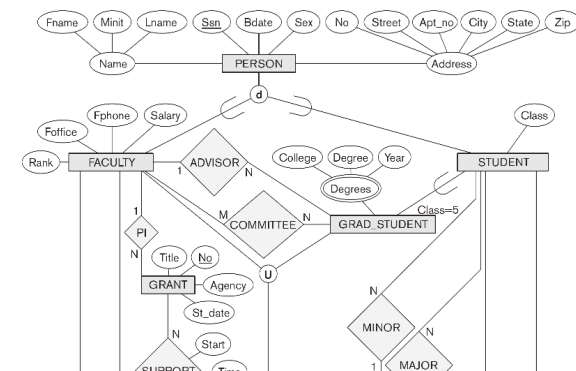


Figure 8.4
Two subclasses (Person
and COMPANY) and
REGISTERED_VEHICLE.

Modeling Union Types

- In the previous example, the OWNER entity is a union type that inherits a set of the attributes of the BANK, COMPANY and PERSON entities depending on the subclass.
- Participation can be *total* or *partial*.
 - Total participation indicates that a subclass holds the union of all entities in its superclass
 - Partial Participation indicates that the subclass can hold a subset of the superclasses
 - Total participation is denoted using the double line between the subclass and the circle.

UNIVERSITY Example



Design Choices

- Conceptual database design is an iterative process of refinement. Some guidelines include:
 - Only represent subclasses where necessary, otherwise the model will become cluttered.
 - Subclasses can be merged into a superclass if they have no specific relationships and few local attributes
 - Avoid using union types unless the situation requires them - try to use specialisation and generalisation as they simplify the model
 - The choice of constraints (e.g. disjoint, overlapping and total, partial) is driven by the mini-world. The default will usually be overlapping-partial.

29 / 34

Summary

- ER Model vs EER Model
- EER Model - Object Oriented Approach
- Specialisation
- Generalisation
- Constraints of Specialisation and Generalisation
- Hierarchies and Lattices
- Modeling Union Types
- UNIVERSITY Example
- Design Choices

30 / 34

Questions?

31 / 34

Next Lecture

- Generating an SQL schema from a ER/EER model..

32 / 34

Reading

- Chapters 4 from *Fundamentals of Database Systems*
- Chapter 7 from *Fundamentals of Database Systems* for Fridays prac session.