Extensions

- existence dependent (weak) entity sets
  - assumption so far: entities exist autonomously and can be uniquely identified within an entity set by their key attributes (strong entity set)
  - in reality there are also weak entities that do not have sufficient attributes to form a key. These entities are
    + dependent in their existence from another, superior entity and
    + can be uniquely identified only in combination with the key of a superior entity
  - superior entity set is called identifying or owner entity set

- graphical notation: 

- identifying relationship set
  - a weak entity set $E_1$ must be associated with an identifying entity set $E_2$ by an identifying relationship set, if the key of $E_1$ comprises the key of $E_2$ and if it contains one or more additional attributes of $E_1$
  - relationship from the weak entity set to the superior entity set has usually an $m:1$-cardinality and more seldom a $1:1$-cardinality

- graphical notation: 
example:

- total participation of an entity set in a relationship
  - all entities of an entity set $E_1$ are associated with another entity set $E_2$ by a relationship set $R$
  - this holds, in particular, for weak entity sets
  - example:

- more precise characterization of cardinalities of relationship sets
  - $(min, max)$-notation
  - for each entity set participating in a relationship set
    - $min$ expresses that each entity of this set is in relationship at least $min$ times
    - $max$ expresses that each entity of this set is in relationship at most $max$ times
- special cases
  + $\textit{min} = 0$: an entity does not have to be in relationship (optional)
  + $\textit{max} = *$: an entity may be in relationship arbitrarily many times
- example: conceptual university schema with $(\textit{min}, \textit{max})$-notations
- **multivalued attributes**
  - optional attribute: minimal cardinality is equal to 0
  - simple attribute: cardinality is equal to 1
  - prescribed attribute: minimal cardinality is equal to 1
  - **multivalued attribute**: maximal cardinality is equal to n
  - example:

![Diagram of multivalued attributes]

- **composite** attributes
  - grouping of attributes of the same entity set or relationship set which are closely related
  - antonym: simple attribute
  - example:

![Diagram of composite attributes]
- derived attributes
  - attribute that can be derived from one or more attributes
  - antonym: base/stored attribute
  - graphical representation:
  - example:

```
  person
      /\  
    /   
  name  birth-date
     \  /
     age
```
Generalization

- goals
  - abstraction at the set level: better (i.e., more understandable and more concise) structuring of entity sets
  - abstraction at the instance level: similar entities are to be modeled by a common entity set

- „factoring“ (extracting) properties (attributes, relationships) of similar entity sets (sub-class, subtypes, categories) to a common superclass (supertype)

- properties that cannot be extracted remain with the respective subclass, i.e., the subclass is a specialization of the superclass

- inheritance as the key concept of generalization: a subclass inherits all properties of a superclass
entities of a subclass are implicitly considered as entities of the superclass, therefore **is-a** in the graphical representation

→ set of entities of the subclass is a subset of the set of entities of the superclass

- **two special cases**
  - **disjoint/overlapping specialization**: all subclasses of a superclass are pairwise disjoint/overlapping
  - **total specialization**: the superclass does not contain explicit elements, but is only given by the union of its subclasses (antonym: **partial specialization**)

![Diagram showing the relationships between entities such as students, employees, assistants, professors, research area, rank, and room.](image-url)
Aggregation

- goal: distinct entity sets which together form a structured superclass are associated with each other
- an aggregation is a special relationship set which associates each superior entity set with several subordinate entity sets
- part-of-relationship
- example: construction of a bicycle
4. Relational Data Model

4.1 Introduction


- commercial DBMSs like Oracle, Informix, SQL Server, Sybase, DB/2 are based on the relational model

- reasons for the success of the relational data model
  - flat tables (relations) as the simple underlying data structure
  - no nested complicated structures
  - set oriented processing of data in contrast to record oriented processing prevailing until then (hierarchical model, network model)
  - simple comprehensibility also for the unskilled user
  - good performance for standard database applications
  - existence of a mature, formal theory (in contrast to other data models), in particular with respect to the design of relational databases and with respect to an efficient processing of user queries