Relationship

- a relationship describes a connection between several entities, e.g., student Smith attends lecture COP 4720, teaching assistant Benson works for professor Meyer
- a homogeneous set of relationships is collected in a relationship set, e.g., relationship sets attends_lecture or works_for
- formal: relationship set $R$ between the entity sets $E_1, E_2, ..., E_n$ as a relation, i.e.,
  $$R \subseteq E_1 \times E_2 \times ... \times E_n,$$
  $n$ degree of relationship set $R$
  - attends_lecture $\subseteq$ students $\times$ lectures
  - works_for $\subseteq$ TAs $\times$ professors
- attributes may characterize relationships, e.g.
  - frequency as an attribute for attends_lecture
- an entity set can occur more than once in a relationship set
- if there is only one entity set $E$ participating in a binary relationship $R(E, E)$, each of these entity sets can be assigned roles
  - e.g., is_precondition_of $\subseteq$ lectures $\times$ lectures
  - first lecture / second lecture has the role of a predecessor / successor
Constraints of binary relationship sets

- **1:1-relationship** (one-to-one relationship)
  - if for a binary relationship set $R(E_1, E_2)$ each entity in $E_1$ is associated with at most one entity in $E_2$, and vice versa

- **1:m-relationship** (one-to-many relationship)
  - if for a binary relationship set $R(E_1, E_2)$ each entity in $E_1$ is associated with any number (zero or more) of entities in $E_2$, and each entity in $E_2$ is associated with at most one entity in $E_1$

- **m:1-relationship** (many-to-one relationship)
  - analogous to the 1:m-relationship

- **m:n-relationship** (many-to-many relationship)
  - if for a binary relationship set $R(E_1, E_2)$ each entity in $E_1$ is associated with any number (zero or more) of entities in $E_2$, and vice versa

- constraints considered as *partial functions*, e.g.
  - for 1:1-relationship: has_husband: women $\rightarrow$ husbands, has_wife: men $\rightarrow$ wives
  - for m:1-relationships: employed_by: persons $\rightarrow$ companies
E-R diagrams

- Graphical representation of entity sets, relationship sets, and their attributes by means of a graph

Notations

- Rectangles represent entity sets: $E$

- Ellipses represent attributes: $A$
  - They are connected with their entity set by undirected edges
  - Key attributes are underlined

- Relationship sets are represented by diamonds: $R$
  - Relationship sets are connected with their pertaining entity sets by edges
  - Edges carry information about cardinality according to imposed constraints

- A role of a relationship set is attached to the corresponding edge
Example: conceptual university schema

```
reg-id | name | sem
students m | attends n | lectures

- grade
- tests

predecessor m | successor n

is_precondition_of

pers-id | name | credits | title
professors 1 | works_for 1 | assistants m

pers-id | name | room

pers-id | rank | room
```
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:

```
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:

```
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
  + min = 0: an entity does not have to be in relationship (optional)
  + max = *: an entity may be in relationship arbitrarily many times
- example: conceptual university schema with (min, max)-notations

```
reg-id | name | sem |
students | attends | lectures
| grade | tests | gives |
| | | |
| | | |
| | | |
| assistants | works_for | professors
| pers-id | pers-id |
| | name |
| | room |
| | |
| | |
| | |
| | |
| | |
| id | credits | title |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
```

- **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:

```
<table>
<thead>
<tr>
<th></th>
<th>first-name</th>
<th>(1,n)</th>
<th>last-name</th>
<th>(0,1)</th>
<th>driving-licence-no</th>
</tr>
</thead>
<tbody>
<tr>
<td>person</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0,n)</td>
<td>phone-no</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

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

```
<table>
<thead>
<tr>
<th></th>
<th>name</th>
<th>street</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>zipcode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>city</td>
</tr>
<tr>
<td>person</td>
<td>birth-date</td>
<td>address</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

- derived attributes
  - attribute that can be derived from one or more attributes
  - antonym: base/stored attribute
  - graphical representation:
  - example:

![Graphical representation](image-url)
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 (subclass, 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**).
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