Features of relations

- no order on the tuples in a relation
  - a relation is defined as a set of tuples, i.e., the tuples in a relation are *not* ordered
  - but: in a file all data records are physically ordered
  - also: the rows in a table are ordered

- order on the values in a tuple and alternative definition of a relation
  - according to the definition of a relation a tuple is an ordered list of \( n \) values
  - From a logical perspective an order of the attributes and their values is not important. It is only necessary to maintain the correspondence between attributes and their values.

- alternative definition of a relation
  - relation schema \( R(A_1, A_2, \ldots, A_n) \) is a set of attributes
  - relation instance \( r_R \) is a finite set of *mappings* \( r_R = \{t_1, t_2, \ldots, t_m\} \), where \( t_i : R \rightarrow D \) mit \( D = \text{dom}(A_1) \cup \text{dom}(A_2) \cup \ldots \cup \text{dom}(A_n) \)
    - \( \forall t \in r_R \quad \forall 1 \leq i \leq n : t(A_i) \in \text{dom}(A_i) \)
    - each mapping \( t_i \) is called *tuple*
    - tuple as a *set* of \((\text{attribute}, \text{value})\)-pairs: each pair yields the value of the mapping from an attribute \( A_i \) to a value \( v_i \in \text{dom}(A_i) \)
values in tuples
- each value in a tuple in atomic (indivisible)
- no composite or multivalued attributes allowed
- first normal form
- values of attributes in a tuple can be unknown or not apply to a specific tuple
- use of a special null value for this case

Keys

- analogously to the notion of key in the E-R model
- due to the set property of relations there are no two tuples that have the same combination of values for all their attributes
- Let us assume $R(A_1, A_2, ..., A_n)$, and let $X \subseteq \{A_1, A_2, ..., A_n\}$. $X$ is called key, if the following conditions are fulfilled:
  - uniqueness: for all relation instances $r_R$ of $R$ holds:
    \[ \forall t_1, t_2 \in r_R : t_1[X] = t_2[X] \Rightarrow t_1 = t_2 \]
  - minimality: there is no $Y \subset X$, so that uniqueness is fulfilled
- candidate keys: several possible keys, one of them is selected as the primary key
More notions

- **database schema**: set of relation schemas
- **database**: set of current relation instances

The definitions so far allow instances that cannot exist in reality. Hence, it makes sense to restrict the instances by suitable semantical conditions.

→ **integrity constraints**
3.3 Transformation of an E-R Schema into a Relational Schema

Data structures

- of the E-R model
  - entity sets
  - relationship sets
- of the relational model
  - relation (schemas)

Problem: How can an E-R data model be transferred into a relational model?

Transformation of a strong entity set

- For each strong entity set $E$ an independent relation schema $R$ is created which comprises all simple attributes of $E$. From a composite attribute only the simple component attributes are taken.
- The names of attributes are generally selected according to the names of properties of the entity set
- The key of the entity set becomes the primary key of the relation schema
- Example: conceptual university schema (repeated)
students(reg-id: integer, name: string, sem: integer)
lectures(id: integer, credits: integer, title: string)
professors(pers-id: integer, name: string, rank: string, room: integer)
assistants(pers-id: integer, name: string, room: integer)
Transformation of a weak entity set

- For each weak entity set $W$ with the respective strong entity set $E$, an independent relation schema $R$ is created which comprises all simple attributes and all simple components of composite attributes of $W$ as attributes of $R$.
- In addition, all primary key attributes of $E$ are added to $R$ as foreign key attributes. The primary key of $R$ then arises from the combination of the primary key of $E$ and the partial key of $W$, if the latter one exists.

Transformation of a 1:1-relationship set

- For each binary 1:1-relationship set $R$ let $S$ and $T$ be the relation schemas that correspond to the entity sets participating in $R$. One of the relation schemas, let us say $S$, is selected, and the primary key of $T$ is added to $S$ as foreign key. It is advantageous to select an entity set with total participation in $R$ for $S$. In addition, all simple attributes and all simple components of composite attributes of $R$ are taken as attributes of $S$.
- Example:
Transformation of a 1:m- and a m:1-relationship set

- For each binary 1:m-relationship set $R$ let $S$ be the relation schema which corresponds to the entity set participating in $R$ on the $m$-side. Add to $S$ as foreign key the primary key of relation schema $T$, which corresponds to the other entity set participating in $R$. The reason for this is that each entity on the $m$-side is associated with at most one entity on the 1-side of $R$. Furthermore, all simple attributes and all simple components of composite attributes of $R$ are taken as attributes of $S$.

- example university database:
  - lectures(id, credits, title, held_by)
  - professors(pers-id, name, room, rank)
  - assistants(pers-id, name, room, boss)

- The names of attributes of a foreign key have partially to be changed in order to ensure the uniqueness of names in a schema.
Transformation of an $m:n$-relationship set

- For each binary $m:n$-relationship set $R$ a new relation schema $S$ is created. Add to $S$ as foreign keys the primary keys of the relation schemas that correspond to the two entity sets participating in $R$. Their combination forms the primary key of $S$. Furthermore, all simple attributes and all simple components of composite attributes of $R$ are taken as attributes of $S$.

- example university database:
  
  `attends(reg-id, id)`

  `is_precondition_of(predecessor, successor)`

Transformation of multivalued attributes

- For each multivalued attribute $A$ a new relation schema $R$ is created. $R$ comprises an attribute corresponding to $A$ and as foreign key the primary key $K$ of the relation schema which corresponds to the entity set or relationship set containing $A$ as attribute. The primary key of $R$ is the combination of $A$ and $K$. If the multivalued attribute is composite, its simple components are added to $R$. $K$ does not contain an attribute corresponding to $A$.

- example:

  ![Diagram](image)

  `department(dept-no, name)`

  `dept-loc(location, dept-no)`