

3. Das Relationale Datenmodell

- ❑ geht zurück auf Codd (1970):
E. F. Codd: A Relational Model of Data for Large Shared Data Banks. Comm. of the ACM 13(6): 377-387(1970)
- ❑ DBMS wie z. B. Oracle, SQL Server, Sybase, Informix, DB/2 basieren auf dem relationalen Modell
- ❑ Relationales Modell
 - Datentypen:
Menge, Multimenge, ...
 - Operationen:
Vereinigung, Differenz, ...

3.1 Basiskomponenten des Datenmodells

- Wertebereich (“domain”): *Integer*, *String*[20], *Datum*, ...
- Relation R ist definiert auf einem **Relationenschema** RS_R .
 - RS_R entspricht einer Menge von Attributen $\{A_1, \dots, A_k\}$
 - zu jedem Attribut A_j gibt es einen Wertebereich $D_j = \text{dom}(A_j)$
 - eine **Instanz** I_R **einer Relation** R ist eine Teilmenge des kartesischen Produkts der Wertebereiche:

$$I_R \subseteq D_1 \times D_2 \dots \times D_k, \quad k \geq 1$$

- Sprechweisen
 - Der Begriff Relation wird oft für die Instanz einer Relation verwendet.
 - Ist klar, um welche Relation es sich handelt, wird der Index bei der Instanz und beim Schema einfach weggelassen.

- Beispiel:

Wertebereiche: $D_1 = \{a, b, c\}$, $D_2 = \{0, 1\}$

Kartesisches Produkt: $D_1 \times D_2 = \{(a, 0), (a, 1), (b, 0), (b, 1), (c, 0), (c, 1)\}$

mögliche Instanzen: $I_1 = \{(a, 0), (b, 0), (c, 0), (c, 1)\}$ und $I_2 = \emptyset$

Notation:

- ❑ **Tupel:** Element einer Relationeninstanz.
- ❑ Für eine Relation mit k Attributen bezeichnet k den **Grad** der Relation.
- ❑ Beschränkung auf endliche Instanzen einer Relation.
- ❑ Darstellung der Relationen durch **Tabellen:**

| R_1 | A | B |
|-------|---|---|
| | a | 0 |
| | b | 0 |
| | c | 0 |
| | c | 1 |

A und B sind hier Spaltennamen (Attributsnamen)

Reihenfolge der Attribute

- ❑ Reihenfolge der Zeilen (Tupel) ist bisher von Bedeutung
- ❑ Wunsch: Reihenfolge der Spalten (Attribute) soll nicht von Bedeutung sein.
 - $D = \cup_{1 \leq j \leq k} \text{dom}(A_j)$
 - Eine Relationeninstanz I ist eine endliche **Menge von totalen Abbildungen**
 $I = \{t_1, \dots, t_n \mid \text{mit } t_j: RS \rightarrow D; t_j(A_m) \in \text{dom}(A_m), 1 \leq m \leq k\}$

Beispiel:

| Städte | <i>Name</i> | <i>Einwohner</i> | <i>Land</i> |
|--------|-------------|------------------|-------------|
| | München | 1.211.617 | Bayern |
| | Bremen | 535.058 | Bremen |

- ❑ Relationenschema der Relation Städte:
 {Name, Einwohner, Land} mit $\text{dom}(\text{Name}) = \text{String}[40]$, $\text{dom}(\text{Einwohner}) = \text{INTEGER}$ und $\text{dom}(\text{Land}) = \text{String}[40]$
- ❑ Ausprägungen: t_1 und t_2
 $t_1(\text{Name}) = \text{München}$, $t_1(\text{Einwohner}) = 1211617$ und $t_1(\text{Land}) = \text{Bayern}$
 $t_2(\text{Name}) = \text{Bremen}$, $t_2(\text{Einwohner}) = 535058$ und $t_2(\text{Land}) = \text{Bremen}$
- ❑ Sei RS ein Relationenschema und $X \subseteq RS$ dann bezeichnet $t[X]$ das Tupel t eingeschränkt auf X .

Notation:

- ❑ **Datenbankschema:** Menge der Relationenschemata
- ❑ Datenbank: Menge der aktuellen Instanzen der Relationen
- ❑ Die bisherige Definition einer Relation läßt Instanzen zu, die real nicht existieren können. Es ist hier sinnvoll, die Instanzen durch geeignete semantische Bedingungen einzuschränken.

Schlüssel

- ❑ analog zum Schlüsselbegriff des ER Modells
- ❑ gegeben: Relation R mit Schema RS
- ❑ $X \subseteq RS$ wird als Schlüssel bezeichnet, wenn folgende Bedingungen erfüllt sind:
 - Eindeutigkeit: für alle (real möglichen) Relationeninstanzen I der Relation R gilt: Für zwei beliebige Tupel t_1 und t_2 aus I gilt:

$$t_1[X] = t_2[X] \Rightarrow t_1 = t_2$$

- Minimalität: es gibt kein $Y \subset X$ ($Y \neq X$), so daß die Eindeutigkeit erfüllt ist.

Beachte:

Schlüsseleigenschaft bezieht sich nicht auf die aktuell abgespeicherten Instanz einer Relation, sondern ist eine semantische Eigenschaft des Relationenschema

- ❑ Mehrere Teilmengen eines Relationenschemas können Schlüssel sein. Dann wird einer von diesen als **Primärschlüssel** ausgezeichnet.
- ❑ Die Attribute des Primärschlüssels werden im Schema oft durch Unterstreichen hervorgehoben.
 - Relation Städte mit Schema {Name, Einwohner, Land}

3.2 Abbildung eines ER Schema in ein relationales Schema

- ❑ Datenstruktur der ER-Datenmodellierung
 - Entitätstypen
 - Beziehungstypen
- ❑ Datenstruktur des relationalen Modells
 - Relationen (bzw. Relationenschema)

Fragestellung

- ❑ Wie kann ein ER Datenmodell in ein relationales Model umgesetzt werden?
- ❑ Diese Frage wird nun in zwei Schritten beantwortet:
 - Einfache Umsetzung von Entitätstypen und Beziehungstypen
 - Konsolidierung des Relationenschema

Umsetzung eines Entitätstyps

- ❑ Der Entitätstyp wird als eigenständige Relation umgesetzt, wobei jede Eigenschaft des Entitätstyps durch ein Attribut in der Relation dargestellt wird.
- ❑ Die Namen der Attribute können entsprechend den Namen der Eigenschaften des Entitätstyps gewählt werden.
- ❑ Der Schlüssel des Entitätstyps wird als Primärschlüssel des Relationenschema vereinbart.
- ❑ Beispiel (siehe ER Diagramm aus Kapitel):
 - Entität Zug:
{ZugNr: *Integer*, Typ: *String*}
 - Entität Bahnhof
{Id: *Integer*, Stadt: *String*, Name: *String*}

Umsetzung eines Beziehungstyps

- ❑ Der Beziehungstyp wird als eigenständige Relation umgesetzt, wobei
 - die Primärschlüssel der beteiligten Entitäten vollständig im Relationenschema aufgenommen werden
 - und jede Eigenschaft des Beziehungstyps durch ein Attribut in der Relation dargestellt wird.
- ❑ Die aus einem Primärschlüssel gewonnen Attribute werden als **Fremdschlüssel** bezeichnet.
- ❑ Die Namen der Attribute eines Fremdschlüssel müssen z. T. umbenannt werden, damit eine Eindeutigkeit der Namen gewährleistet wird.
- ❑ Beispiel (m:n Beziehungstyp)
 - Umsetzung des m:n Beziehungstyps *hält* und die an dem Beziehungstyp beteiligten Entitätstypen *Zug* und *Bahnhof*.
 - Der Schlüssel von *hält* entspricht dabei der Menge von Fremdschlüsseln.

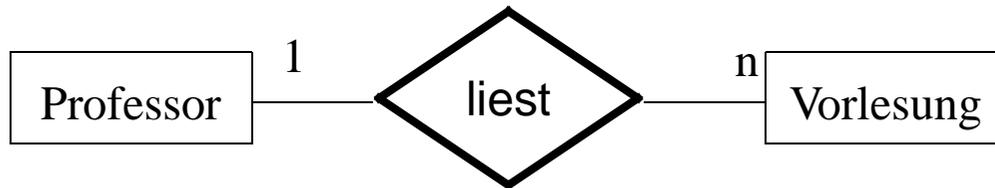
| <i>Zug</i> | <u>Id</u> | <i>Typ</i> |
|------------|-----------|------------|
| | 2475 | IR |
| | 95786 | RB |
| | ... | ... |

| <i>hält</i> | <u>Zug</u> | <i>Station</i> | <i>Gleis</i> |
|-------------|------------|----------------|--------------|
| | 2475 | 1 | 4 |
| | 2475 | 2 | 10 |
| | 95786 | 1 | 2 |
| | ... | ... | ... |

| <i>Bahnhof</i> | <u>Id</u> | <i>Stadt</i> | <i>Name</i> |
|----------------|-----------|--------------|-------------|
| | 1 | Marburg | Lahn |
| | 2 | Darmstadt | Hbf |
| | 3 | Wetter | Hess-Nass |
| | ... | ... | |

□ Beispiel (Umsetzung eines 1:n Beziehungstyps)

- Betrachten wir hierzu ein Beispiel aus der Universitätswelt, wo ein Professor Vorlesungen abhält. Wenn der Primärschlüssel von *Professor* die *PersNr* und die von *Vorlesung* die *VorNr* ist, würden die Beziehung *liest* als zweistellige Relation umgesetzt werden.



- Man beachte aber, daß $\{PersNr, VorNr\}$ kein Schlüssel von *liest* ist, da die Eigenschaft der Minimalität verletzt ist. $\{VorNr\}$ erfüllt aber die Eigenschaft eines Schlüssels.

□ Mit der Fremdschlüsseleigenschaft bezeichnet man nun folgende Nebenbedingung:

- Sei F der Fremdschlüssel in R , der sich auf eine Relation S bezieht, so muß stets

$$I_R[F] \subseteq I_S[F]$$

erfüllt sein.

Konsolidierung des Schemas

- ❑ Mögliche Vereinfachungen des Schemas durch Verschmelzen von Relationen, die aus einer zweistelligen Beziehungstyp der Kardinalität 1:1 oder 1:n hervorgegangen sind.
- ❑ Seien R und S, die Relationen der beteiligten Entitätstypen eines 1:n Beziehungstyps, und T die Relation des Beziehungstyps. Dann können wir alternativ folgendes Schema verwenden:
 - Statt der Relation S und T wird eine neue Relation U eingeführt, wobei

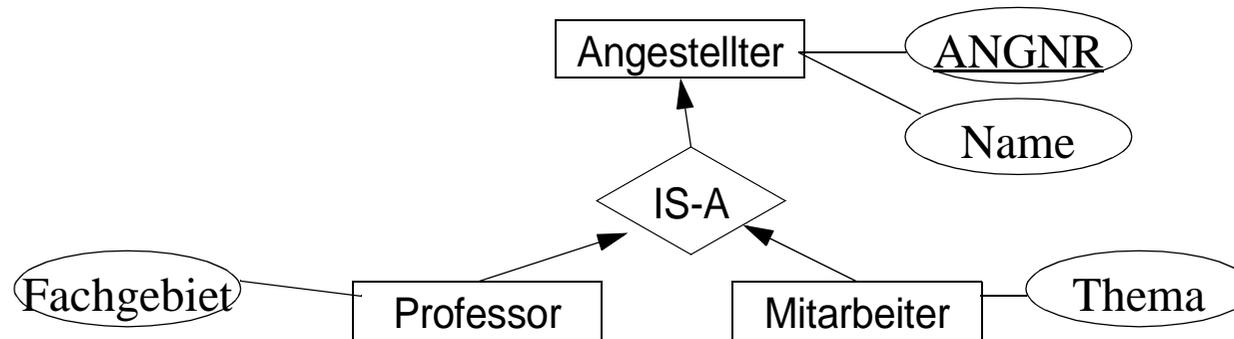
$$RS_U = RS_S \cup RS_T$$

Die Instanzen von U sollen nun die gleiche Information wie zuvor die Instanzen von S und T enthalten. Hierzu verknüpfen wir die Tupel $s \in I_S$ und $t \in I_T$ bzgl. ihres Schlüssels F. Falls $s[F] = t[F]$, wird ein Tupel $u \in I_U$ erzeugt, so daß $u[RS_S] = s$ und $u[RS_T] = t$. Für die Tupel $s \in I_S$, die kein $t \in I_T$ mit $s[F] = t[F]$ besitzen, wird ein Tupel u erzeugt, so daß $u[RS_S] = s$. Die anderen Werte von u werden mit einem speziellen Wert (NULL) aufgefüllt.

- Die Relation R bleibt unverändert bestehen.
- ❑ Was sind die Vor- und Nachteile einer solchen Konsolidierung?

Umsetzung von IS-A Beziehungstypen

- ❑ IS-A Beziehungstypen werden nicht durch eine eigene Relation repräsentiert. Die Beziehung wird dadurch bereits ausgedrückt, wenn der Schlüssel des allgemeinen Typs auch als Schlüssel der spezialisierten Typen benutzt wird.
- ❑ Beispiel:



Daraus ergeben sich drei Relationen mit dem Schema $\{\underline{\text{ANGNR}}, \text{Name}\}$, $\{\underline{\text{ANGNR}}, \text{Fachgebiet}\}$ und $\{\underline{\text{ANGNR}}, \text{Thema}\}$.

- ❑ Als Alternative bietet sich auch an, den IS-A Beziehungstyp und die beteiligten Entitätstypen durch eine Relation mit Schema $\{\underline{\text{ANGNR}}, \text{Name}, \text{Fachgebiet}, \text{Thema}\}$ zu präsentieren und bei den Instanzen die fehlende Information durch NULL-Werte aufzufüllen.

Was ist bei diesem Ansatz problematisch?