Informationssysteme 1 -
Das relationale Modell

Inhalt

Allgemeines

Das Relationale Modell wurde Anfang der 70-ger Jahre konzipiert. Als Erfinder gilt **Codd**, der in seinem Artikel 1970 das relationale Modell erstmals vorstellte.

Das relationale Datenmodell ist im Vergleich zu seinen Vorgängern (Hierarchisches Modell, Netzwerkmodell), die satzorientiert arbeiteten, sehr einfach strukturiert.

Es gibt im Wesentlichen nur flache Tabellen (Relationen), die durch entsprechende Operatoren ausschließlich **mengenorientiert** verarbeitet und verknüpft werden.

Die Einfachheit des Modells, aber auch die fundierte theoretische Ausformulierung, sind wesentliche Gründe für die marktdominierende Stellung der relationalen Datenbanktechnologie.

Definition des relationalen Modells

Mathematischer Formalismus [1]

Gegeben sind n unterschiedliche **Wertebereiche** (**Domains**, **Domänen**):

$D_1, D_2, \ldots, D_n$ ($D_i = D_j$ *ist zulässig für* $i \neq j$)

Diese Domains dürfen nur **atomare Werte** enthalten. Typische Domains sind z.B. Zahlen, Zeichenketten, etc., wobei z.B. Records, Mengen und Listen keine zulässigen Wertebereiche sind. (Datumsobjekte sind als Domain erlaubt.)

Eine **Relation** R **ist** dann definiert als eine **Teilmenge des kartesischen Produkts der** n **Domains**:

$$R \subseteq D_1 \times D_2 \times \ldots \times D_n$$

Ein Element der Menge R wird als Tupel bezeichnet, dessen Stelligkeit (arity) gleich n ist.

Mathematischer Formalismus [2]

Entsprechend dem vorgestellten (mathematischen) Formalismus sieht eine Telefonbuchrelation folgenderweise aus:

$$Telefonbuch \subseteq string \times string \times integer$$

In vielen Systemen werden Relationen auch als **Tabellen** (**Tables**) bezeichnet, weil sie sich leicht als flache Tabelle darstellen lassen.

Beisp:
(entnommen aus [1])

Telefonbuch		
Name	**Straße**	**Telefon#**
Mickey Mouse	Main Street	4711
Minnie Mouse	Broadway	94725
Donald Duck	Broadway	95672
...	...	...

Die Spalten werden als **Attribute** (Felder) bezeichnet. Ihr Name muss innerhalb einer Relation eindeutig sein. Die Zeilen der Tabelle entsprechen dann den Tupeln der Relation.

Mathematischer Formalismus [3]

Ein **Relationenschema** wird wie folgt spezifiziert (Beisp. Telefonbuch):

Telefonbuch: {[Name: string, Adresse: string, <u>TelefonNr.: integer</u>]}

- Attribute und deren Domains werden in eckigen Klammern angegeben.
- Primärschlüsselattribute werden unterstrichen.
- Die geschwungene Klammer soll symbolisieren, dass es sich bei der Ausprägung um eine Menge von Tupeln handelt.

Die eckigen Klammern stellen somit den Tupelkonstruktor, die geschwungenen den Mengenkonstruktor dar. Eine Relation ist also auch hier eine Menge von Tupeln.

In der Tabellendarstellung (siehe vorige Folie) ist das Schema aus dem Tabellennamen und den Spaltenköpfen ersichtlich.

Definition des relationalen Modells

Mathematischer Formalismus [4]

Die **Ausprägung** entspricht dann den einzelnen Tupeln der Relation, bzw. Zeilen in der Tabelle.

$t \in R$ Beisp.: t = („Mickey Mouse“, „Main Street“, 4711)

Soll eine Trennung von Schema und Ausprägung im Formalismus explizit gemacht werden, wird folgende Notation eingesetzt:

sch(R) oder *R* .... die Menge der Attribute (Schema)

dom(A) .... Domain des Attributs *A*

$$R = \{A_1, \ldots A_n\} \qquad R \subseteq \mathrm{dom}(A_1) \times \mathrm{dom}(A_2) \times \ldots \times \mathrm{dom}(A_n)$$

Allgemeines [1]

Das ER-Schema besitzt 2 grundlegende Strukturkonzepte:

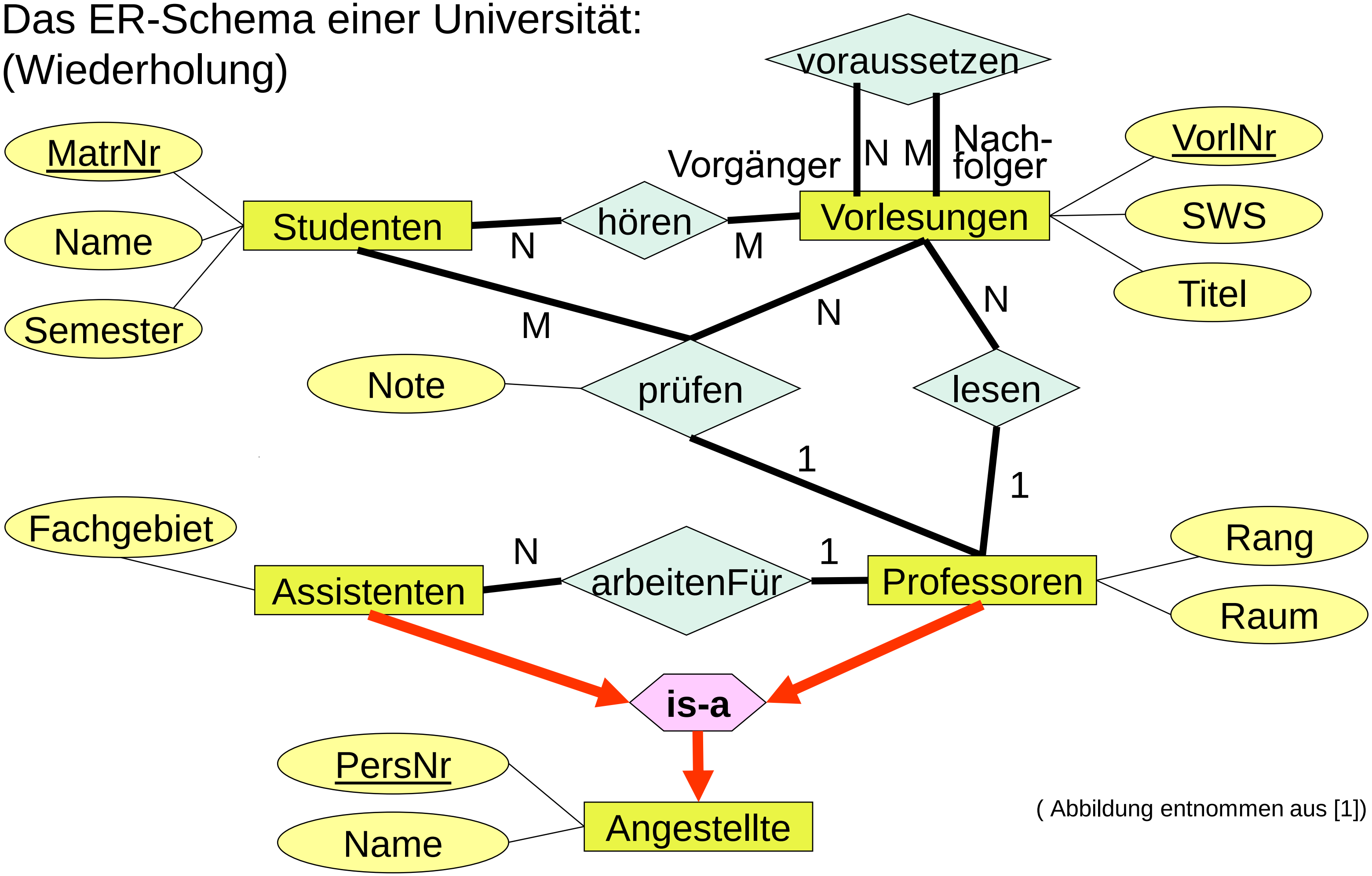
- Entitytypen
- Beziehungstypen

Das relationale Modell hat nur ein einziges Konzept, die Relation. Also müssen sowohl Entitytypen als auch Beziehungstypen jeweils auf Relationen abgebildet werden.

Dasselbe gilt dann auch für die Generalisierung und Aggregation.

Allgemeines [2]

Das ER-Schema einer Universität:
(Wiederholung)



(Abbildung entnommen aus [1])

Entitytypen

Entitytypen werden direkt in Relationen abgebildet:

Beispiel aus dem ER-Diagramm einer Universität:

Studenten : { [<u>MatrNr : integer</u>; Name : string; Semester : integer] }

Vorlesungen : { [<u>VorlNr : integer</u>; Titel : string; SWS : integer] }

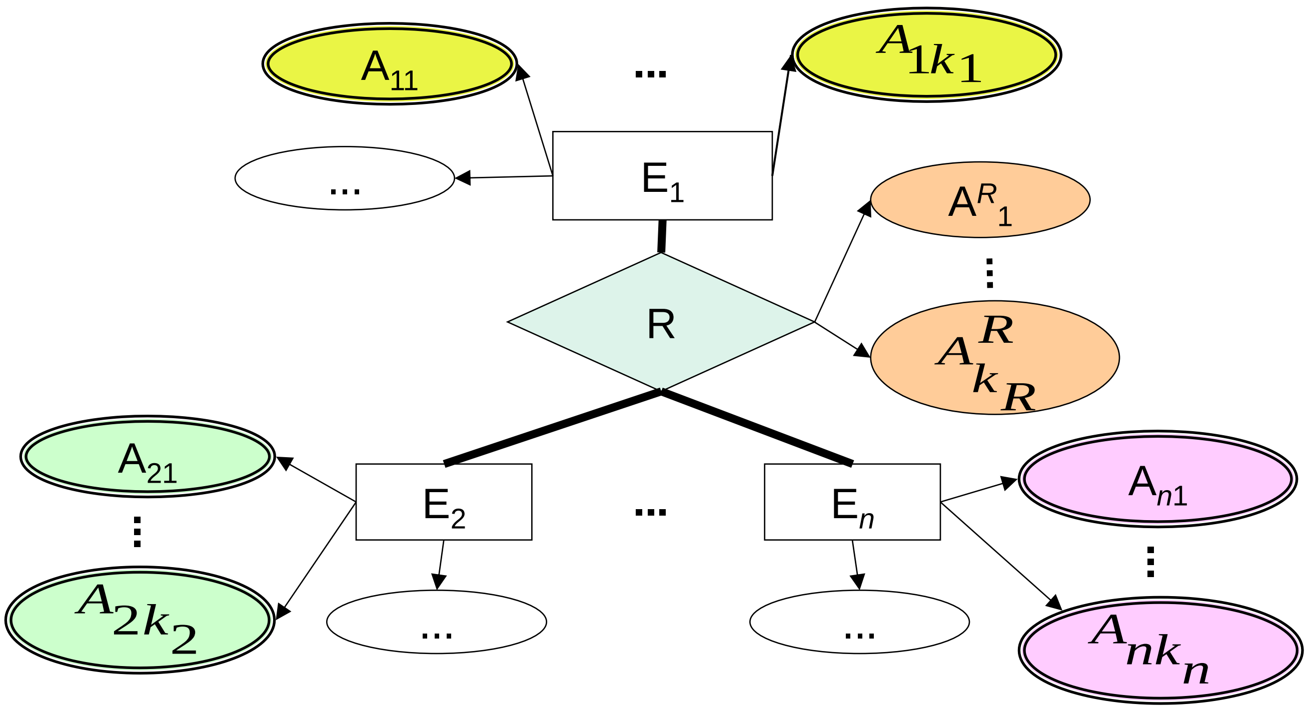
Professoren : { [<u>PersNr : integer</u>; Name : string; Rang : string; Raum : integer] }

Assistenten : { [<u>PersNr : integer</u>; Name : string; Fachgebiet : string] }

Beziehungstypen [1]

Allgemein, Abstrakte n-stellige Beziehung [1] **:**

Im ER-Modell:



Beziehungstypen [2]

Allgemein, Abstrakte n-stellige Beziehung [2] **:**

Im relationalen Model:

$$R:\{[\underbrace{A_{11},\dots,A_{1k_1}}_{\text{Schlüssel von } E_1},\underbrace{A_{21},\dots,A_{2k_2}}_{\text{Schlüssel von } E_2},\dots,\underbrace{A_{n1},\dots,A_{nk_n}}_{\text{Schlüssel von } E_n},\underbrace{A_1^R,\dots,A_{k_R}^R}_{\text{Attribute von } R}]\}$$

Die Relation R enthält also alle Schlüsselattribute der an dem Beziehungstyp beteiligten Entitytypen und zusätzlich die Attribute des Beziehungstyps selbst.

Die Schlüsselattribute der Entitytypen nennt man Fremdschlüssel (Foreign Keys), weil sie Tupel aus anderen Relationen identifizieren.

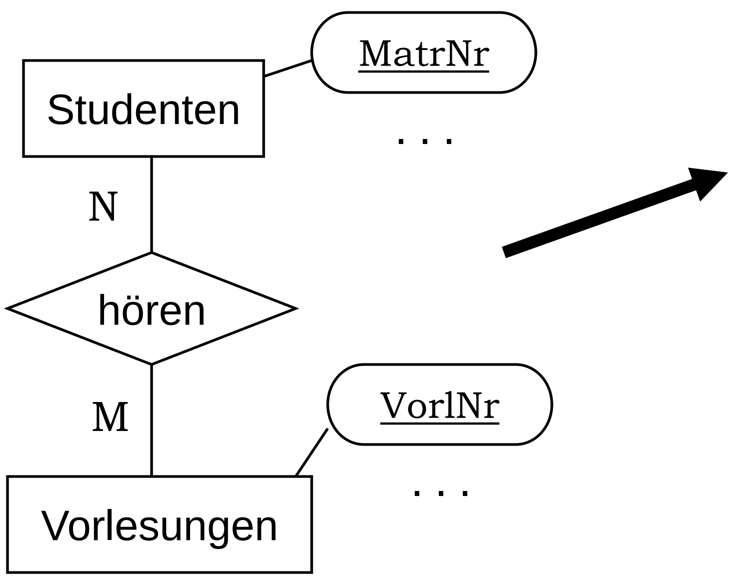
Es kann notwendig sein, dass einige aus den Entitytypen übernommenen Attribute umbenannt werden müssen, weil ja innerhalb einer Relation der Attributname eindeutig zu sein hat.

Beziehungstypen [3]

N:M-Beziehungen:

Sie werden wie die allgemeinen n-stelligen Beziehungen behandelt.

Beisp.:



Studenten : { [MatrNr; Namel; Semester] }
Vorlesungen : { [VorlNr; Titel; SWS; gelesenVon] }
hören : { [MatrNr; VorlNr] }

Studenten	
MatrNr	**...**
26120	...
27550	...
...	...

hören	
MatrNr	**VorlNr**
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022
29555	5001

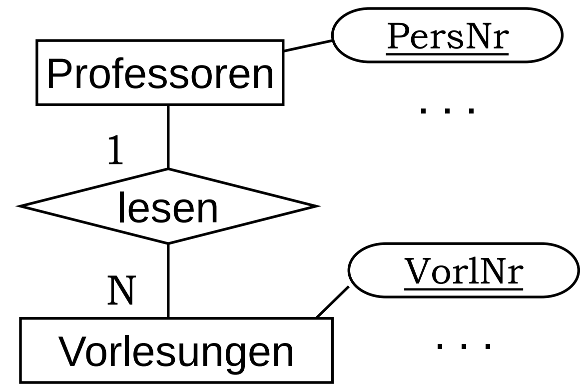
Vorlesungen	
VorlNr	**...**
5001	...
4052	...
...	...

Beziehungstypen [4]

1:N-Beziehungen:

Primärschlüssel der „1-Seite“ wird Fremdschlüssel auf der „N-Seite“.

Beisp.:



Vorlesungen			
VorlNr	Titel	SWS	Gelesen Von
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

Vorlesungen : { [VorlNr; Titel; SWS; gelesenVon] }
Professoren : { [PersNr; Name; Rang; Raum] }

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

Das Attribut ‚PersNr‘ wurde in der Relation ‚Vorlesung‘ auf ‚gelesenVon‘ umbenannt (wegen besserer Lesbarkeit)

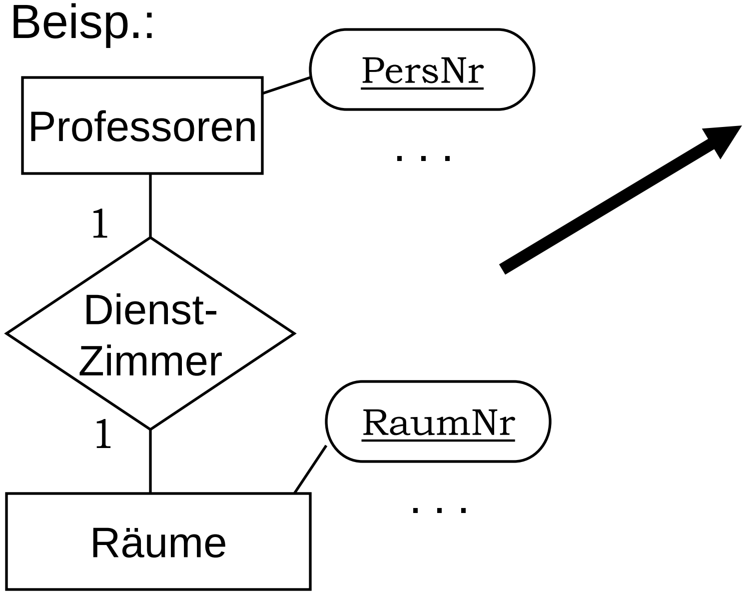
Beziehungstypen [5]

1:1-Beziehungen:

Sie können wie 1:N-Beziehungen behandelt werden.

Es ist jedoch nicht gleichwertig, in welche ‘Richtung‘ modelliert wird. Es ist jene Variante zu bevorzugen, bei der weniger sogenannte ‚NULL-Einträge‘ entstehen.

Beisp.:



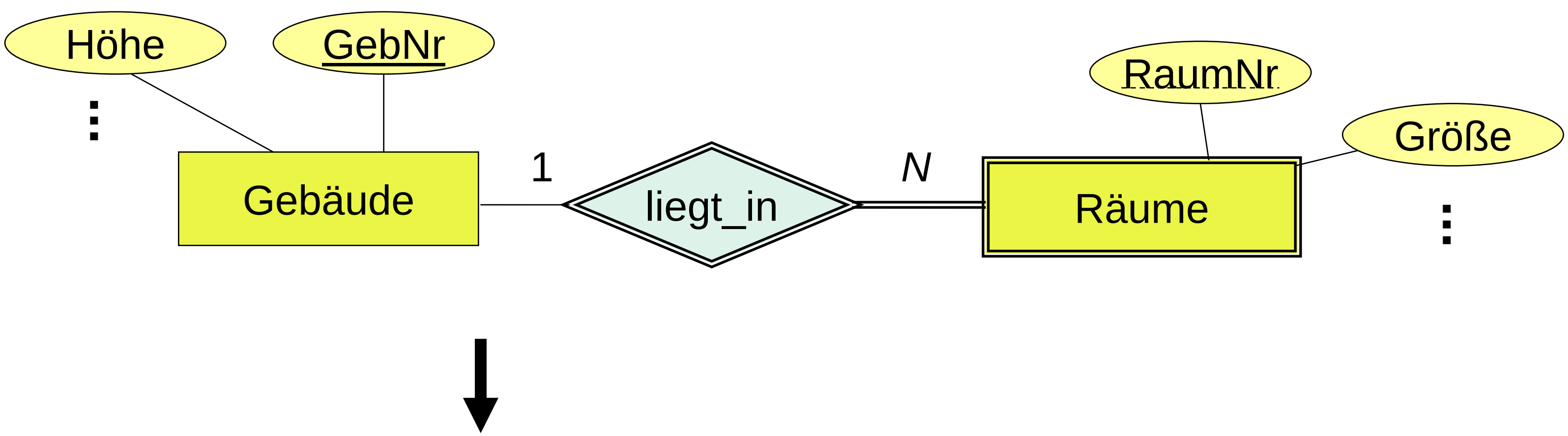
Professoren : { [PersNr; Name; Rang; Raum] }
Räume : { [RaumNr; Größe; Lage] }

Entitytypen mit gleichem Primärschlüssel können (müssen nicht) zu einer Relation zusammengefasst werden.

Schwache Entitytypen

Der Primärschlüssel des Entitytps, von dem der schwache Entititytyp abhängig ist, wird Teil des Primärschlüssels der Relation, die den schwachen Entitytyp darstellt.

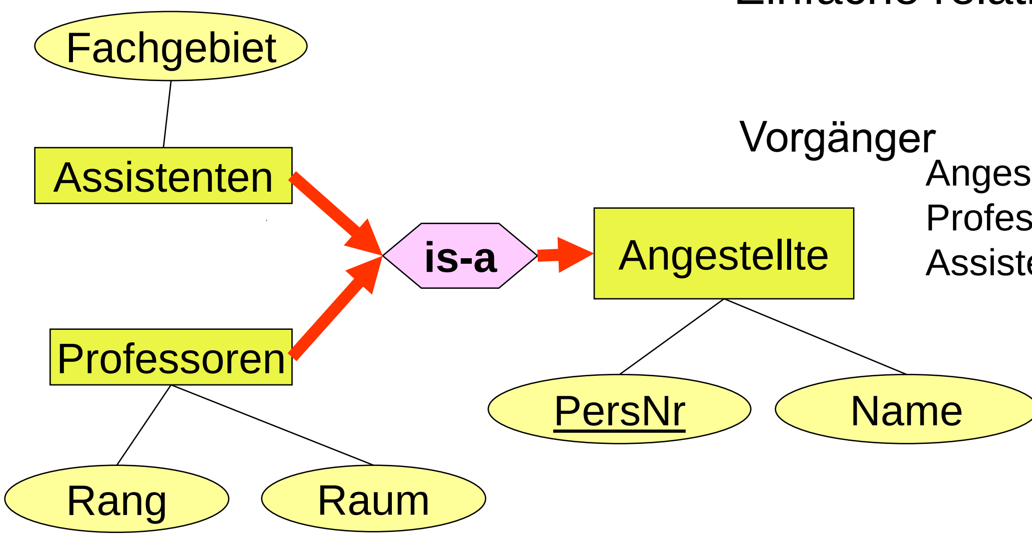
Beisp.:



Gebäude : { [<u>GebNr</u>; Höhe; . . .] }
Räume : { [<u>GebNr</u>; <u>RaumNr</u>; Größe; . . .] }

Generalisierung

Beisp.:



Einfache relationale Repräsentation:

Vorgänger

Angestellte : { [PersNr; Name] }
Professoren : { [PersNr; Rang; Raum] }
Assistenten : { [PersNr; Fachgebiet] }

Diese Tansformation hat den Nachteil, dass in den Relationen, die die Spezialisierung darstellen, nicht die volle Information verfügbar ist. Man muss sie erst mit der Relation, die den Obertyp realisiert, verbinden. Die Vererbung ist somit nicht im relationalen Modell realisiert. Dieser Nachteil kann zum Teil über Views (siehe bei SQL) wettgemacht werden.

(Abbildung entnommen aus [1])

Aggregation

Die Aggregation wird (genau genommen wie die Generalisation) vom relationalen Datenmodell nicht unterstützt.

Jeder ‚part-of'-Beziehungstyp ist entsprechend der Regeln für Beziehungstypen (1:N, 1:1) in das relationale Modell zu überführen.

Beispiel

Die relationale Universitäts-DB:

(Ohne der Generalisation ‚Angestellte‘)

Assistenten			
PersNr	**Name**	**Fachgebiet**	**Boss**
3002	Platon	Ideenlehre	2125
3003	Aristoteles	Syllogistik	2125
3004	Wittgenstein	Sprachtheorie	2126
3005	Rhetikus	Planetenbewegung	2127
3006	Newton	Keplersche Gesetze	2127
3007	Spinoza	Gott und Natur	2126

Vorlesungen			
VorlNr	**Titel**	**SWS**	**gelesen von**
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

hören	
MatrNr	**VorlNr**
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022

prüfen			
MatrNr	**VorlNr**	**PersNr**	**Note**
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2

Studenten		
MatrNr	**Name**	**Semester**
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555	Feuerbach	2

Professoren			
PersNr	**Name**	**Rang**	**Raum**
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

voraussetzen	
Vorgänger	**Nachfolger**
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

Allgemeines

Es gibt zwei formale Sprachen, die für die Abfrageformulierung in relationale Datenbanken konzipiert wurden:

- **Die Relationale Algebra:**
  Sie ist eher prozedural orientiert. D.h. ein Ausdruck beinhaltet implizit einen Abarbeitungsplan für die Abfrage. Deshalb spielt sie eine größere Rolle bei der Realisierung von Datenbanksystemen, insbesondere auch bei der Abfrageoptimierung.
- **Der Relationenkalkül:**
  Er ist rein deklarativ. D.h. es wird nur spezifiziert, welche Kriterien die Daten erfüllen müssen, die man aus der Datenbank erhalten will.

Beide Sprachen sind **abgeschlossen**. D.h., die Ergebnisse der Abfragen sind wiederum Relationen.

Die relationale Algebra [1]

Selektion:

Es werden jene Tupel ausgewählt, die das **Selektionsprädikat** erfüllen. Sie wird mit σ bezeichnet und hat das Selektionsprädikat als Subskript.

Beisp.: $\sigma_{Semester>10}$ (Studenten)

$\sigma_{Semester > 10}$ (Studenten)		
MatrNr	**Name**	**Semester**
24002	Xenokrates	18
25403	Jonas	12

Allgemein ist das Selektionsprädikat eine Formel F, die aufgebaut ist aus
- Attributnamen der Argumentrelation R oder Konstanten als Operanden
- den Vergleichsoperatoren (z.B.:$=,<,\leq,>,\geq,\neq$) und
- den logischen Operatoren $\wedge$, $\vee$, $\neg$.

(Entsprechende Klammer-Setzung ist erlaubt.)

Dann besteht das Ergebnis der Selektion σ_F (R) aus allen Tupeln $t \in R$, für die die Formel F erfüllt ist, wenn jedes Vorkommen eines Attributnamens A in F durch den Wert t.A ersetzt wird.

Die relationale Algebra [2]

Projektion:

Während bei der Selektion Tupel (Zeilen) einer Relation (Tabelle) ausgewählt werden, werden bei der Projektion Attribute (Spalten) extrahiert. Sie wird mit dem Symbol π bezeichnet und enthält als Subskript die Menge der ausgewählten Attributnamen.

Beisp.: π_{Rang} (Professoren)

Π_{Rang}**(Professoren)**
Rang
C4
C3

Achtung: Bei der Projektion werden Duplikattupel, die durch die Enschränkung der Attributmenge entstehen können, eliminiert.

Die relationale Algebra [3]

Vereinigung:

Zwei Relationen mit gleichem Schema (mit gleichen Attributnamen und Domains) kann man mittels Vereinigung zu einer Relation zusammenfassen.

Beisp.: $\pi_{\text{PrsNr,Name}}$ (Assistenten) $\cup$ $\pi_{\text{PrsNr,Name}}$ (Professoren)

Achtung: Auch bei der Vereinigung werden Duplikattupel eliminiert.

Mengendifferenz:

Für zwei Relationen R und S ist die Mengendifferenz R - S definiert als die Menge der Tupel, die in R aber nicht in S vorkommen.

Beisp.: π_{MatrNr} (Studenten) - π_{MatrNr} (prüfen)

Die relationale Algebra [4]

Kartesisches Produkt (Kreuzprodukt):

Das Kreuzprodukt zweier Relationen R und S $(R \times S)$ enthält alle möglichen Paare von Tupeln aus R und S. Es gilt:

$$\text{sch}(R \times S) = \text{sch}(R) \cup \text{sch}(S)$$

Beisp.: Professoren $\times$ hören

Professoren				hören	
PersNr	**Name**	**Rang**	**Raum**	**MatrNr**	**VorlNr**
2125	Sokrates	C4	226	26120	5001
...	...	...	...	...	...
2125	Sokrates	C4	226	29555	5001
...	...	...	...	...	...
2137	Kant	C4	7	29555	5001

Achtung: Würden bei einem Kreuzprodukt gleiche Attributnamen entstehen, werden diese mit dem entsprechenden Relationennamen qualifiziert (**qualifizierte Attributnamen**). Schreibweise: R.A (Beisp.: Assistenten.PersNr)

Die relationale Algebra [5]

Umbenennung von Relationen und Attributen [1]**:**

Wird eine Relation mehrfach in einer Abfrage verwendet, ist es notwendig, (zumindest logisch) eine vollständige Kopie dieser Relation zu erzeugen und sie umzubenennen. Der Operator dafür ist das ρ, wobei im Subskript der neue Name angegeben wird.

Beisp.: ρ_{V1} (voraussetzen)

Umbenennung von Attributen:

Beisp.: $\rho_{\text{Voraussetzung} \leftarrow \text{Vorgänger}}$ (voraussetzen)

(Attribut ‚Vorgänger‘ der Relation ‚voraussetzen‘ wird auf ‚Voraussetzung‘ umbenannt.)

<u>Komplexes Beispiel:</u> „Ermittlung indirekter Vorgänger 2. Stufe der Vorlesung 5216“

$$\pi_{\text{V1.Vorgänger}}(\sigma_{\text{V2.Nachfolger=5216} \wedge \text{V1.Nachfolger V2.Vorgänger}}(\rho_{\text{V1}}(\text{voraussetzen}) \times \rho_{\text{V2}}(\text{voraussetzen})))$$

Die relationale Algebra [6]

Definition der relationalen Algebra:

Die bislang eingeführten Operatoren sind ausreichend, um die relationale Algebra formal definieren zu können:

Basisausdrücke:
- Relationen der Datenbank
- konstante Relationen

Sind E_1 und E_2 relationale Ausdrücke (Basisausdrücke oder bereits komplexere, zusammengesetzte Ausdrücke) dann sind die folgenden Ausdrücke auch gültige Algebraausdrücke:

- $E_1 \cup E_2$, wobei $sch(E_1) = sch(E_2)$ gelten muss
- $E_1 - E_2$, wiederum Schemagleichheit vorausgesetzt
- $E_1 \times E_2$
- $\sigma_P (E_1)$, mit einem Prädikat P über den Attributen in E_1
- $\pi_S (E_1)$, mit $S \subseteq sch(E_1)$
- $\rho_V (E_1)$ und $\rho_{A \leftarrow B} (E_1)$, wobei $B \in sch(E_1)$ und $A \notin sch(E_1)$

Die in den folgenden Folien vorgestellten Operatoren, lassen sich alle durch diese Operatoren ausdrücken!

Die relationale Algebra [7]

Der relationale Verbund (Join) [1]

Der natürliche Verbund [1]

Wenn die Relationen R insgesamt $m+k$ Attribute $A_1, \ldots, A_m, B_1, \ldots, B_k$ und die Relation S $n+k$ Attribute $B_1, \ldots, B_k, C_1, \ldots, C_n$ dann ist $R \mid\times\mid S$ wie folgt definiert.

$$R \mid\times\mid S = \pi_{A_1,\ldots,A_m,R.B_1,\ldots,R.B_k,C_1,\ldots,C_n}\left(\sigma_{R.B_1=S.B_1 \wedge \ldots \wedge R.B_k=S.B_k}(R \times S)\right)$$

R ⋈ S											
R – S				**R ∩ S**				**S – R**			
A_1	A_2	...	A_m	B_1	B_2	...	B_k	C_1	C_2	...	C_n

Es wird also das Kreuzprodukt gebildet, aus dem dann nur jene Tupel selektiert werden, deren Attributwerte für gleichbenannte Attribute der beiden Argumentrelationen gleich sind. Weiters werden diese gleichbenannten Attribute nur einmal ins Ergebnis übernommen.

Die relationale Algebra [8]

Der relationale Verbund (Join) [2]

Der natürliche Verbund [2]

Beispiel: Die Daten der Studenten, kombiniert mit den Daten der Vorlesungen, die sie hören.

(Studenten |×| hören) |×| Vorlesungen

(Studenten ⋈ hören) ⋈ Vorlesungen						
MatrNr	**Name**	**Semester**	**VorlNr**	**Titel**	**SWS**	**gelesenVon**
26120	Fichte	10	5001	Grundzüge	4	2137
27550	Jonas	12	5022	Glaube und Wissen	2	2134
28106	Carnap	3	4052	Wissenschftstheorie	3	2126
...	...	...	...	...	...	...

Der Join-Operator ist assoziativ und kommutativ. Also könnten im obigen Beispiel die Klammern anders gesetzt (Studenten |×| (hören |×| Vorlesungen) oder auch weggelassen werden (Studenten |×| hören |×| Vorlesungen).

Die relationale Algebra [9]

Der relationale Verbund (Join) [3]

Allgemeiner Join (Theta-Join)

Habe *R* die Attribute $A_1, \ldots, A_n$, *S* die Attribute $B_1, \ldots, B_m$ und sei Attributen θ ein beliebiges Prädikat über diesen Attributen (z.B.: $A_1 > B_1 \wedge A_2 = B_5$):

$R \bowtie_\theta S$							
R				S			
A_1	A_2	...	A_n	B_1	B_2	...	B_m

Der Theta-Join ist also nur eine vereinfachte Formulierung eines Kreuzprodukts, gefolgt von einer Selektion:

$$R \mid\times\mid_\theta S = \sigma_\theta(R \times S)$$

Das Ergebnis hat also n+m Attribute, gleichnamige Attribute werden wiederum qualifiziert. Einen Theta-Join der Form $R \mid\times\mid_{R.Ai = S.Bi} S$ nennt man **Equi-Join**. Im Unterschied zum natürlichen Verbund müssen die Attribute nicht gleich benannt sein und sowohl $R.A_i$ und $S.B_i$ werden ins Ergebnis aufgenommen.

Die relationale Algebra [10]

Der relationale Verbund (Join) [4]

Weitere Join-Operatoren [1]

Die bislang eingeführten Join-Operatoren nennt man auch ‚innere Joins', weil jene Tupel, die keinen „Join-Partner" gefunden haben, nicht in die Ergebnismenge aufgenommen werden.

Äußere Joins (Outher Joins)

- linker äußerer Join (left outher join)]×| :
  Die Tupel der linken Argumentrelation bleiben in jedem Fall erhalten
- rechter äußerer Join (right outher join) |×[:
  Die Tupel der rechten Argumentrelation bleiben in jedem Fall erhalten.
- (vollständiger) äußerer Join (left outher join)]×[:
  Die Tupel beider Argumentrelationen bleiben in jedem Fall erhalten.

Semi Joins

- linker Semi Join |× : $R \mid\!\times S = \pi_{\mathrm{sch}(R)}(R \mid\!\times\!\mid S)$
- rechter Semi Join × | : analog dem linken Semi-Join

Die relationale Algebra [11]

Der relationale Verbund (Join) [5]

Zusammenfassung:

• **natürlicher Join**

L		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$\bowtie$

R		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

=

Resultat				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1

• **linker äußerer Join**

L		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

⟕

R		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

=

Resultat				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_2	-	-

• **rechter äußerer Join**

L		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

⟖

R		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

=

Resultat				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
-	-	c_3	d_2	e_2

• **äußerer Join**

L		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

⟗

R		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

=

Resultat				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_2	-	-
-	-	c_3	d_2	e_2

• **Semi-Join von L mit R**

L		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$\ltimes$

R		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

=

Resultat		
A	B	C
a_1	b_1	c_1

• **Semi-Join von R mit L**

L		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$\rtimes$

R		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

=

Resultat		
C	D	E
c_1	d_1	e_1

Die relationale Algebra [12]

Mengendurchschnitt

$R \cap S = R - (R - S)$ (Schemagleichheit wird wieder vorausgesetzt)

Die relationale Division

Für die Division $R \div S$ muss $\text{sch}(S) \subseteq \text{sch}(R)$ gelten. Das Schema der Ergebnisrelation ist dann $\text{sch}(R) - \text{sch}(S)$. Ein Tupel t ist in $R \div S$ enthalten, wenn es für jedes Tupel t_s aus S ein Tupel t_r aus R gibt, so dass folgende beiden Bedingungen erfüllt sind.

$t_r.\text{sch}(S) = t_s.$ (sch(S) und $t_r.(\text{sch}(R) - \text{sch}(S)) = t$

<u>Beisp.:</u> Die Matrikelnummer jener Studenten, die alle 4-stündigen Vorlesungen gehört haben.

$\text{hören} \div \pi_{\text{VorlNr}}(\sigma_{\text{SWS}=4}(\text{Vorlesungen}))$

R	
M	V
m_1	v_1
m_1	v_2
m_1	v_3
m_2	v_2
m_2	v_3

÷

S
V
v_1
v_2

=

R ÷ S
M
m_1

Darstellung der Division durch andere Operatoren:

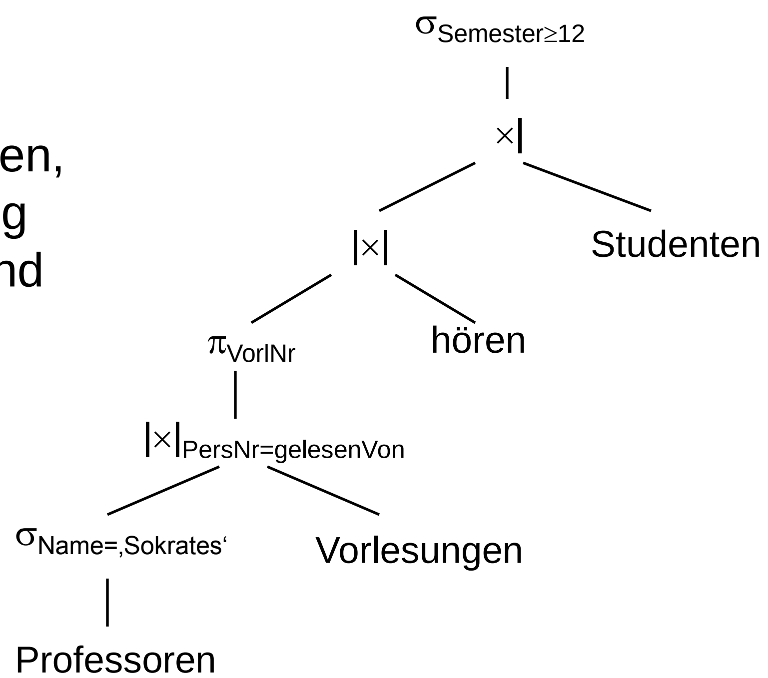
$R \div S = \pi_{\text{sch}(R) - \text{sch}(S)}(R) - \pi_{\text{sch}(R) - \text{sch}(S)}((\pi_{\text{sch}(R) - \text{sch}(S)}(R) \times S) - R)$

Die relationale Algebra [13]

Operatorbaum-Darstellung

Bislang wurden alle Operationen „in-line“ dargestellt. Bei komplexen Anfragen kann eine sogenannte Operatorbaum-Darstellung übersichtlicher sein.

<ins>Beisp.:</ins> Finde all jene Studenten, die mindestens eine Vorlesung von Sokrates gehört haben und schon im 12. oder einem höheren Semester sind.



Der Relationenkalkül [1]

Allgemeines

Der Relationenkalkül basiert auf dem mathematischen Prädikatenkalkül erster Stufe, der quantifizierte Variablen und Werte zulässt. Es gibt zwei unterschiedliche, aber gleich mächtige Ausprägungen des Relationenkalküls:

- der relationale Tupelkalkül
- der relationale Domänenkalkül

Der Unterschied liegt darin, dass die Variablen des Kalküls im einen Fall an Tupel, im anderen Fall an Domänen (Wertemengen von Attributen) gebunden werden.

Jede Anfrage des Tupelkalküls kann ein eine äquivalente Anfrage des Domänenkalküls umformuliert werden, und umgekehrt.

Der Relationenkalkül [2]

Beispielanfrage im relationalen Tupelkalkül

Allgemeine Form: $\{t|P(t)\}$

t ... Tupelvariable, freie Variable des Prädikats P (darf nicht durch einen Existenz- oder Allquantor gebunden sein)

P... Prädikat, das erfüllt sein muss, damit t in das Ergebnis aufgenommen wird

Beisp.: {p | p∈Professoren ∧ p.Rang = ‚C4‘ }

Mit Hilfe des Tupelkonstruktors [$t_1.A_1$, ... , $t_n.A_n$] ist es möglich, neue, noch nicht in der Datenbank existierende Tupel aufzubauen. A_1... A_n sind Attributnamen, die in den Schemata der Relationen in P vorkommen müssen.

Beisp.: {[p.Name,a.PersNr | p∈Professoren ∧ a∈Assistenten ∧ p.PersNr = a.Boss }

Der Relationenkalkül [3]

Quantifizierung von Tupelvariablen

Allgemeine Form: $\exists t \in R(Q(t))$ bzw. $\forall t \in R(Q(t))$

- t ... Tupelvariable, die durch die Quantifizierung an die Relation R gebunden wird
- $Q(t)$... Prädikat, das erfüllt sein muss, damit t in das Ergebnis aufgenommen wird

Beispiele:

- Jene Studenten, die mindestens eine Vorlesung bei ‚Curie' gehört haben:
  {s | s ∈ Studenten
  ∧ ∃h ∈ hören(s.MatrNr=h.MatrNr
  ∧ ∃v ∈ Vorlesungen (h.VorlNr=v.VorlNr
  ∧ ∃p ∈ Professoren (p.PersNr=v.gelesenVon
  ∧ p.Name = ‚Curie')))}
- Jene Studenten, die alle 4-stündigen Vorlesungen gehört haben :
  {s | s ∈ Studenten ∧ ∀v ∈ Vorlesungen (v.SWS=4 ⇒
  ∃h ∈ hören(h.VorlNr=v.VorlNr ∧ h.MatrNr=s.MatrNr))}

Der Relationenkalkül [4]

Formale Definition des Tupelkalküls [1]

Allgemeine Form: $\{v|F(v)\}$

v ist eine freie Tupelvariable. In der Formel F können weitere Tupelvariablen vorkommen, die dann nicht mehr frei sein können. Eine Variable ist frei, falls sie nicht durch einen einen Quantor (∃ oder V) gebunden ist.

Atome:

$s \in R$,wobei s eine Tupelvariable und R ein Relationenname ist.

$s.A \ \phi \ t.B$,wobei s und t Tupelvariablen sind, A und B Attributnamen ϕ und ein Vergleichsoperator ist. Es sind $=,<,\leq,>,\geq$ und $\neq$ als Vergleichsoperatoren erlaubt, wobei sie in den betroffenen Wertebereichen definiert sein müssen.

$s.A \ \phi \ c$,wobei $s.A$ die gleiche Bedeutung hat und c eine Konstante im Wertebereich von A ist.

Der Relationenkalkül [5]

Formale Definition des Tupelkalküls [2]

Formeln:

Sie werden nach folgenden Regeln aufgebaut:

- Alle Atome sind Formeln.
- Falls P eine Formel ist, ist auch $\neg P$ und (P) eine Formel.
- Falls P_1 und P_2 Formeln sind, sind auch $P_1 \wedge P_2$, $P_1 \vee P_2$ und $P_1 \Rightarrow P_2$ Formeln.
- Falls $P(t)$ eine Formel mit der freien Variablen t ist, dann sind auch $\exists t \in R(P(t))$ bzw. $\forall t \in R(P(t))$ Formeln.

Der Relationenkalkül [6]

Sichere Ausdrücke des Tupelkalküls

Ein Ausdruck des Tupelkalküls heißt sicher, wenn das Ergebnis des Ausdrucks eine Teilmenge der Domäne ist.

Eine Domäne enthält alle Werte, die als Kosntante in der Formel vorkommen und alle Werte (Attributwerte in Tupeln) der Relationen, die in der Formel vorkommen (namentlich erwähnt sind).

Beispiele:

$\{p \mid p \in \text{Professoren} \wedge p.\text{Rang} = \text{‚C4‘}\}$	ist sicher
$\{n \mid \neg (n \in \text{Professoren})\}$	ist nicht sicher

Der Relationenkalkül [7]

Der relationale Domänenkalkül [1]

Allgemeine Form: $\{[v_1,\ldots v_n] \mid P(v_1,\ldots v_n)\}$

Hier sind die v_i ($1 \leq i \leq n$) die Variablen (Domänenvariablen), die einen Attributwert repräsentieren. P ist ein Prädikat (bzw. eine Formel) mit den freien Variablen $v_1,\ldots v_n$.

Atome:

$[w_1,\ldots w_m] \in R$	ist ein Atom,wobei R eine m-stellige Relation ist. Die Zuordnung der m Domänenvariablen $w_1,\ldots w_m$ zu den Attributen der Relation R erfolgt nach der Reihenfolge der Attribute im Schema von R.
$x \phi y$	ist ein Atom, wobei x und y Domänenvariablen sind, und ϕ und ein Vergleichsoperator ($=,<,\leq,>,\geq$) ist, der auf die Domäne anwendbar ist.
$x \phi c$	ist ein Atom analog oben, wobei c eine Konstante aus der Domäne von x ist.

Der Relationenkalkül [8]

Der relationale Domänenkalkül [2]

Formeln:

- Alle Atome sind Formeln.
- Falls P eine Formel ist, ist auch $\neg P$ und (P) eine Formel
- Falls P_1 und P_2 Formeln sind, sind auch $P_1 \wedge P_2$, $P_1 \vee P_2$ und $P_1 \Rightarrow P_2$ Formeln.
- Falls $P(v)$ eine Formel mit der freien Variablen v ist, dann sind auch $\exists v\ P(v)$ bzw. $\forall v\ (P(v))$ Formeln.

Beispiel:

- Jene Studenten, die mindestens eine Prüfung bei ‚Curie' abgelegt haben:

  { [m,n] | ∃ s ([m,n,s] ∈ Studenten ∧ ∃ v, p, g ([m, v, p, g] ∈ prüfen ∧
  ∃ a, r, b ([p, a, r, b] ∈ Professoren ∧ a = ‚Curie')))}

Im Domänenkalkül werden Joinbedingungen implizit durch die Verwendung derselben Domänenvariablen spezifiziert. Würde man sie expilzit formulieren:

{ [m,n] | ∃ s ([m,n,s] ∈ Studenten ∧ ∃ m', v, p, g ([m', v, p, g] ∈ prüfen ∧ m=m' ∧
∃ p', a, r, b ([p', p, a, r, b] ∈ Professoren ∧ p=p' ∧ a = ‚Curie')))}

Der Relationenkalkül [9]

Sichere Ausdrücke des Domänenkalkül

Ein Ausdruck $\{[x_1,\ldots, x_n] \mid P(x_1,\ldots, x_n)$ ist sicher, wenn folgende Bedingungen erfüllt sind:

- Falls ein Tupel $[c_1,\ldots, c_n]$ mit Konstante c_i im Ergebnis enthalten ist, so muss c_i ($1 \leq i \leq n$) in der Domäne von P enthalten sein.
- Für jede existenz-quantifizierte Teilformel $\exists x\, P_1(x)$ muss gelten, dass P_1 nur für Elemente aus der Domäne vom P_1 erfüllbar sein kann, oder für gar keine. (Wenn für eine Konstante c das Prädikat P_1 erfüllt ist, so muss c in der Domäne von P_1 enthalten sein.)
- Für jede all-quantifizierte Teilformel $\forall x\, P_1(x)$ muss gelten, dass sie dann und nur dann erfüllt ist, wenn $P_1(x)$ für alle Werte der Domäne von P_1 erfüllt ist. ($P_1(d)$ muss für alle d, die nicht in der Domäne von P_1 enthalten sind, auf jeden Fall erfüllt sein.)

Ausdruckskraft der Abfragesprachen

Codd hat 1972 die Ausdruckskraft der relationalen Abfragesprachen definiert:

Eine Abfragesprache heißt **relational vollständig**, wenn sie mindestens so mächtig ist wie die relationale Algebra bzw. der Relationenkalkül.

Die drei Sprachen

- relationale Algebra,
- relationaler Tupelkalkül und
- relationaler Domänenkalkül

besitzen dieselbe Ausdruckskraft. Es lässt sich jeder Ausdruck einer Sprache äquivalent in der anderen Sprache formulieren.