Datenmodellierung/Datenbanksysteme

VU 184.685/VU 184.686, WS 2022

Das relationale Modell

Johannes Fichte, Felix Winter

Institut für Logic and Computation
Technische Universität Wien

FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

Acknowledgments

Die Folien sind eine Weiterentwicklung der Folien von Sebastian Skritek und Katrin Seyr, welche wiederum auf den zum Lehrbuch zur Verfügung gestellten Folien von A. Kemper basieren.

Der Inhalt basiert auf und behandelt Kapitel 3 des Lehrbuchs (Kemper, Eickler: Datenbanksysteme – Eine Einführung)

Übersicht

- Das Relationale Modell
- Umsetzung eines konzeptuellen in ein logisches Schema
  - "Übersetzung" von EER in relationales Schema
  - Eigenschaften relationaler Schemata
- Datenabfragesprachen
  - Relationale Algebra
  - Relationenkalkül
  - Ausdruckskraft von Abfragesprachen

Das Relationale Modell: Begriffsklärung

Name

Attribute

Tupel (Zeile)

Telefonbuch		
Name: string	Adresse: string	TelNr: integer
Mickey Mouse	Main Street	4711
Minnie Mouse	Broadway	94725
Donald Duck	Highway	95672
...	...	...

Schema

Ausprägung

Feld

Das Relationale Modell: Begriffsklärung

Schema:

Telefonbuch: (Name: string, Adresse: string, Telefonnr: integer)

Ausprägung

{
("Mickey Mouse", "Main Street", 4711),
("Minnie Mouse", "Broadway", 94725),
("Donald Duck", "Highway", 95672)
}

ACHTUNG:
Keine Ordnung!
Keine Duplikate!

Relationes Modell: Begriffsklärung

Definition (Relation)

Seien $D_1, D_2, \ldots, D_n$ Domänen (Wertebereiche).
Eine Relation R über $D_1, D_2, \ldots, D_n$ ist eine Teilmenge des kartesischen Produkts der Domänen:

$$R \subseteq D_1 \times \cdots \times D_n$$

n ... Stelligkeit der Relation

Relationale Datenbank: Menge von Relationen

Relationes Modell: Begriffsklärung

Beispiel

$$R \subseteq \text{string} \times \text{string} \times \text{integer}$$

Mickey Mouse	Main Street	4711
Minnie Mouse	Broadway	94725
Donald Duck	Highway	95672
...	...	...

Relationes Modell: Begriffsklärung

Definition (Tupel)

Ein Tupel t ist ein Element einer Relation R

$$t \in R$$

Beispiel

$$t = (\text{“Mickey Mouse”}, \text{“Main Street”}, 4711)$$

Relationes Modell: Begriffsklärung

Definition (Relationenschema)

Das Schema einer Relation besteht aus

- Name der Relation
- Liste der Attribute

$$\text{RelName}\colon (\text{Attr}_1 : \text{dom}_1, \ldots, \text{Attr}_n : \text{dom}_n)$$

Beispiel

Telefonbuch: (Name: string, Adresse: string, Telefonnr: integer)

Schlüssel

Definition (Schlüssel)

Ein Schlüssel ist eine minimale Menge von Attributen, deren Werte ein Tupel eindeutig identifizieren.

Anmerkung: Im Allgemeinen sind mehrere Schlüssel möglich. Ein Schlüsselkandidat wird als Primärschlüssel ausgewählt (Kennzeichnung durch Unterstreichung).

Beispiel

Telefonbuch: (Name: string, Adresse: string, <u>Telefonnr: integer</u>)
Telefonbuch: (<u>Name: string, Adresse: string</u>, Telefonnr: integer)

Begriffsklärung: Fremdschlüssel

Fremdschlüssel Ein Fremdschlüssel ist eine Menge von Attributen welche auf den Schlüssel einer (anderen) Relation verweist.

Beispiel

Raum: (<u>RaumNr: integer</u>, Bezeichnung: string, Ort: string)

Mitarbeiter: (<u>PNR: int</u>, Name: string,
RaumNr: integer: Raum.RaumNr)

Lernziele

- Was ist / Woraus besteht
  - eine Relation
  - ein Relationenschema
  - ein Schlüssel/Primärschlüssel
  - ein Fremdschlüssel

Umsetzung eines konzeptuellen Schemas in ein logisches Schema

EER $\rightarrow$ relationales Schema

Zusammenfassung: Konzepte

Entitytypen

Attribute

Beziehungstypen

Schwache Entitytypen

Generalisierung

Schlüssel

“schwache Schlüssel”

Kardinalitäten

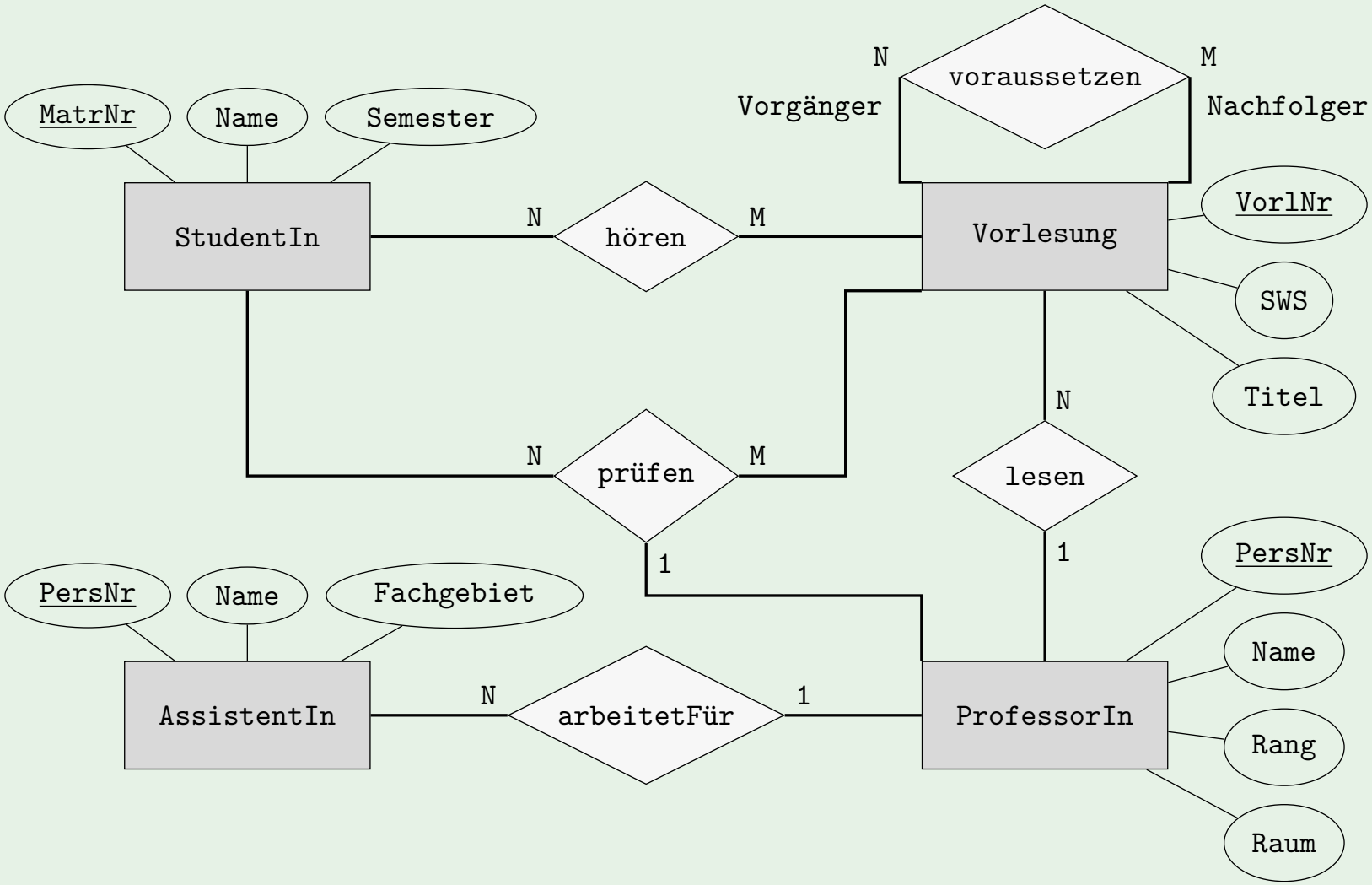
Relationenschema

Schlüssel

Fremdschlüssel

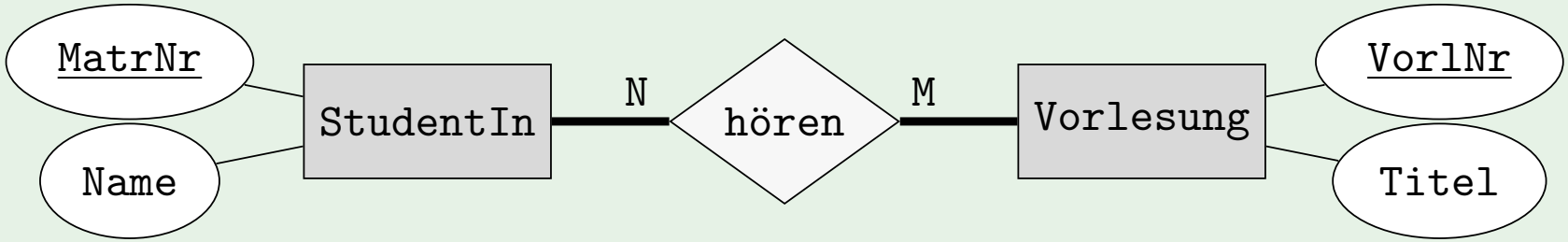
Von EER zum relationalen Schema: Unischema

Beispiel (Unischema)



EER → relationales Schema: Intuition

Beispiel



StudentIn

MatrNr	Name
24002	Xenokrates
25403	Jonas
26120	Fichte
26830	Aristoxenos
...	...

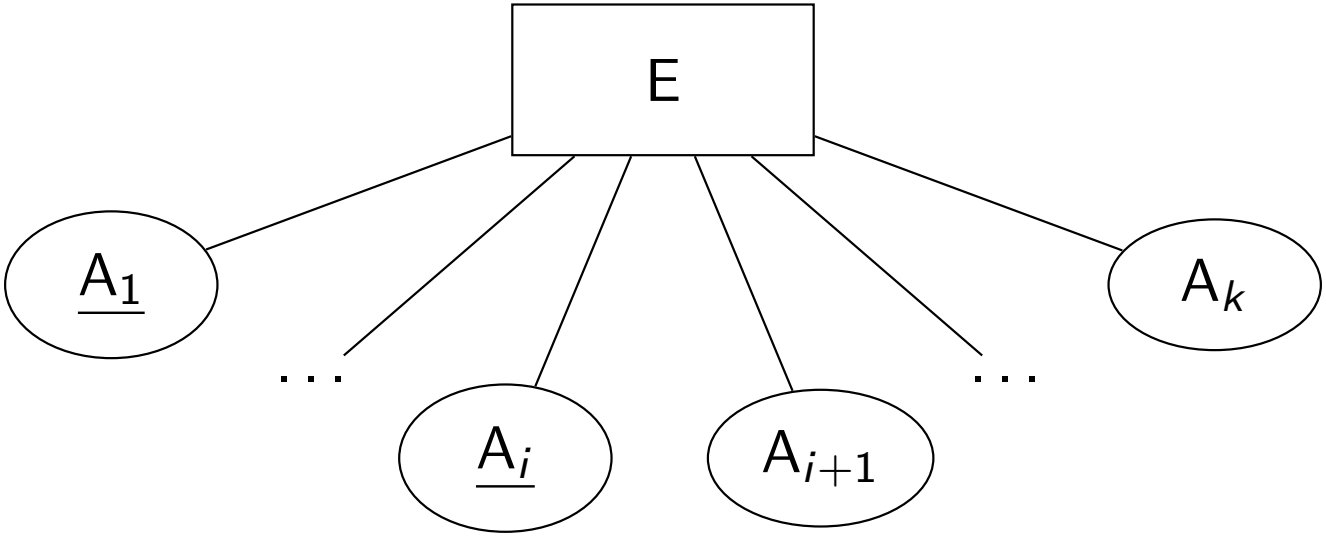
hören

MatrNr	VorlNr
26120	5001
24002	5001
24002	4052
...	...

Vorlesung

VorlNr	Titel
5001	Grundzüge
5041	Ethik
4052	Logik
...	...

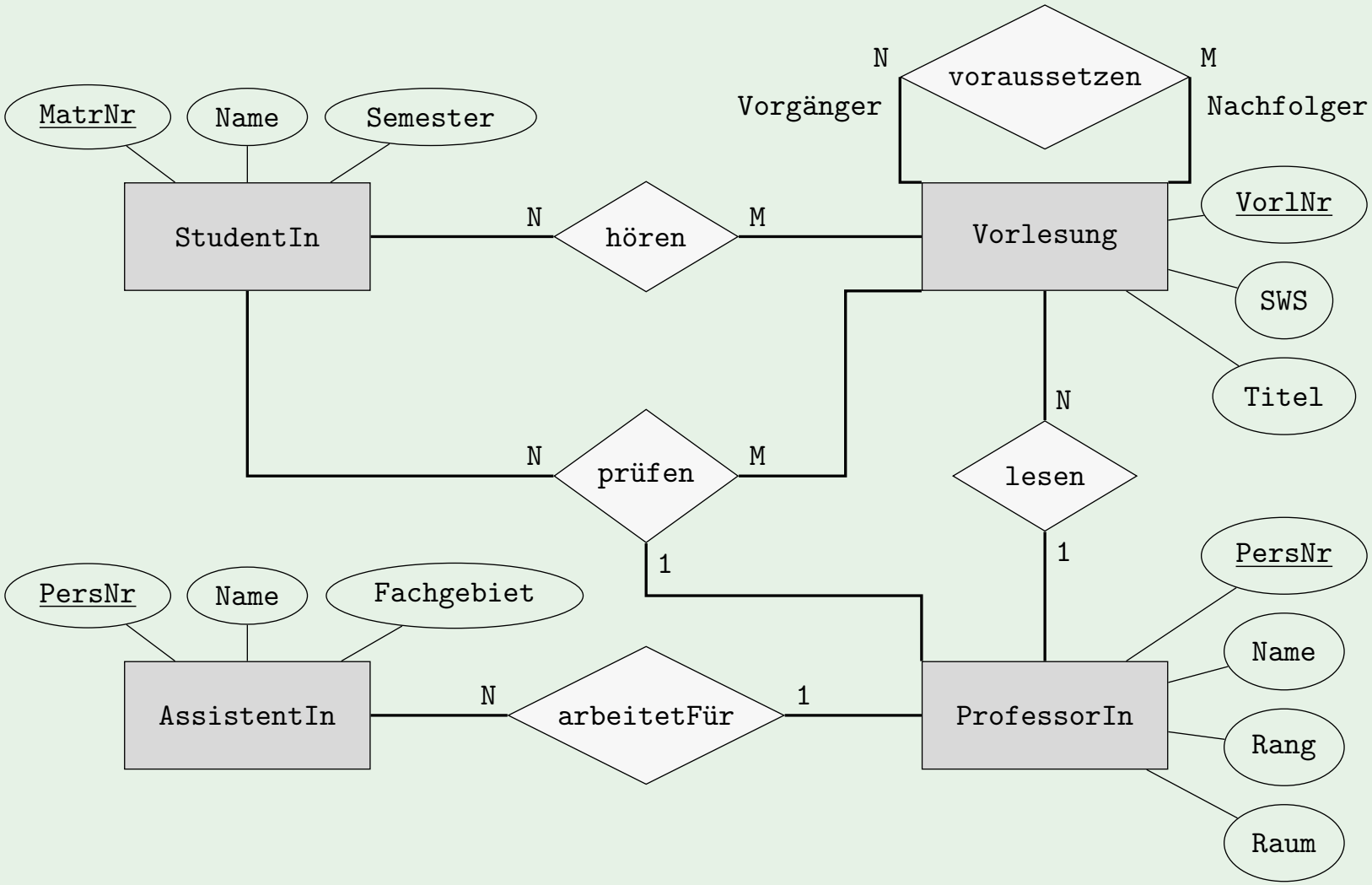
Relationale Darstellung von Entitytypen



$$E\colon\ (\ \underline{A_1 : typ_1,\ \ldots,\ A_i : typ_i},\ A_{i+1} :\ typ_{i+1},\ \ldots,\ A_k :\ typ_k\ )$$

Unischema

Beispiel (Unischema)



Relationale Darstellung von Entitytypen

Relationale Darstellung der vier Entitytypen aus dem Unischema

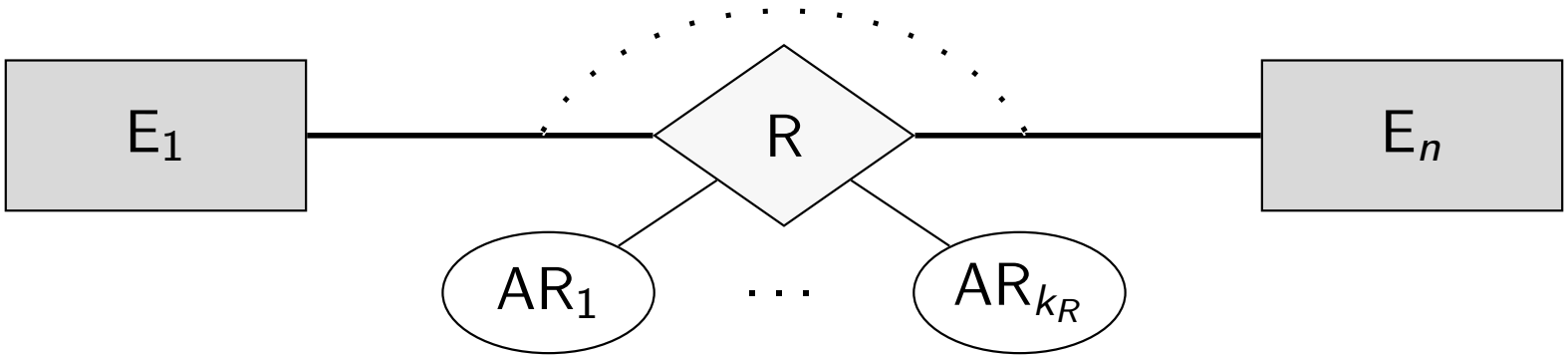
StudentIn: (MatrNr: integer, Name: string, Semester: integer)

Vorlesung: (VorlNr: integer, Titel: string, SWS: integer)

ProfessorIn: (PersNr: integer, Name: string, Rang: string, Raum: integer)

AssistentIn: (PersNr: integer, Name: string, Fachgebiet: string)

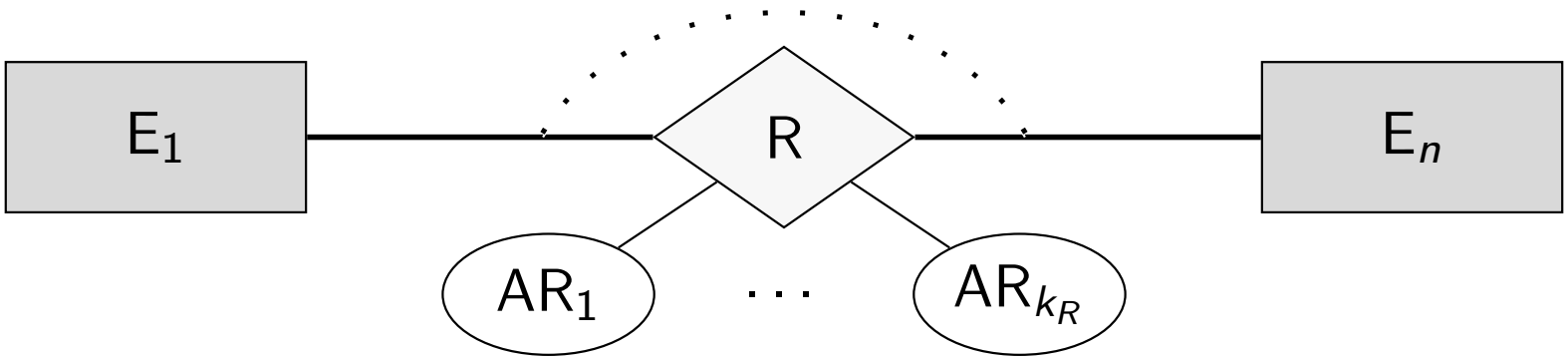
Relationale Darstellung von Beziehungstypen



Intuitiv:

R: (Schlüssel von $rel(E_1)$, ..., Schlüssel von $rel(E_n)$, Attribute von R)

Relationale Darstellung von Beziehungstypen



Annahme: $E_i(\underline{\mathrm{A}^i_1, \ldots, \mathrm{A}^i_{k_i}}, \mathrm{B}^i_1, \ldots \mathrm{B}^i_{\ell_i})$ (für alle $1 \leq i \leq n$)

$R\colon\ (\ A^1_1 : E_1.A^1_1, \ldots, A^1_{k_1} : E_1.A^1_{k_1},$ Schlüssel von $rel(\mathrm{E}_1)$

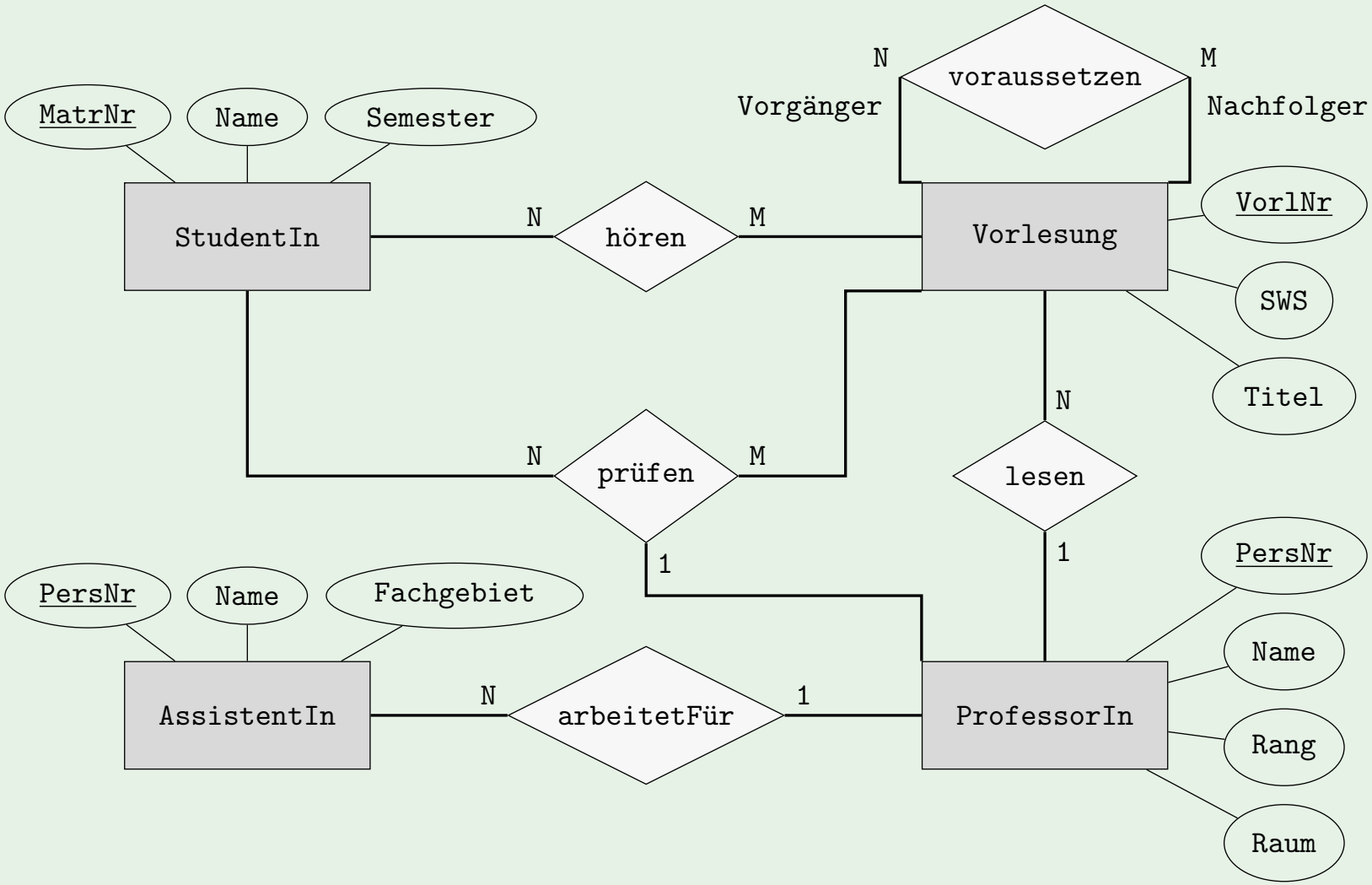
$\ldots,$

$A^n_1 : E_n.A^n_1, \ldots, A^n_{k_n} : E_n.A^n_{k_n},$ Schlüssel von $rel(\mathrm{E}_n)$

$\mathrm{AR}_1, \ldots, \mathrm{AR}_{k_R})$ Attribute von R

Von EER zum relationalen Schema: Unischema

Beispiel (Unischema)



Relationale Darstellung von Beziehungstypen

Relationale Darstellung der Beziehungstypen aus dem Unischema (Fremdschlüssel implizit)

hören: (*MatrNr: integer*, *VorlNr: integer*) (N:M)

lesen: (*PersNr: integer*, *VorlNr: integer*) (1:N)

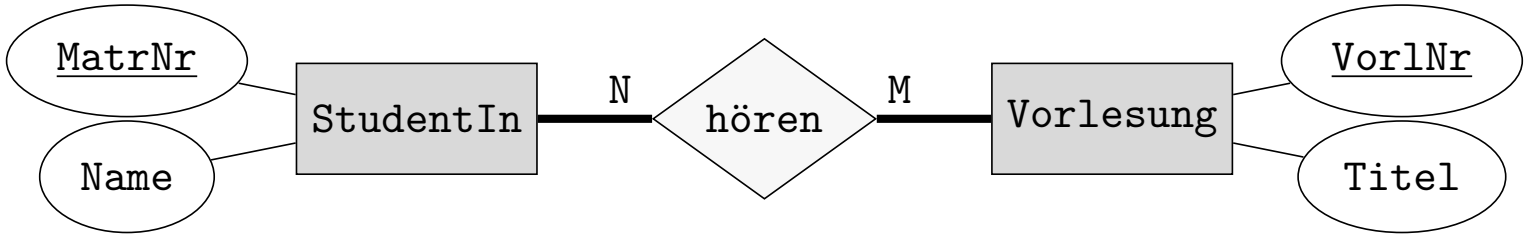
arbeitenFür: (*AssiPersNr: integer*, *ProfPersNr: integer*) (N:1)

voraussetzen: (*Vorgänger: integer*, *Nachfolger: integer*) (N:M)

prüfen: (*MatrNr: integer*, *VorlNr: integer*, *PersNr: integer*, Note: decimal) (N:M:1)

Schlüssel?

N:M Beziehung (hören)



StudentIn	
MatrNr	Name
24002	Xenokrates
25403	Jonas
26120	Fichte
26830	Aristoxenos
28106	Carnap
29555	Feuerbach
...	...

hören	
MatrNr	VorlNr
24002	4052
24002	5001
26120	5001
...	...

Vorlesung	
VorlNr	Titel
5001	Grundzüge
5041	Ethik
5049	Mäeutik
4052	Logik
5216	Bioethik
...	...

Schlüssel von N:M Beziehungen

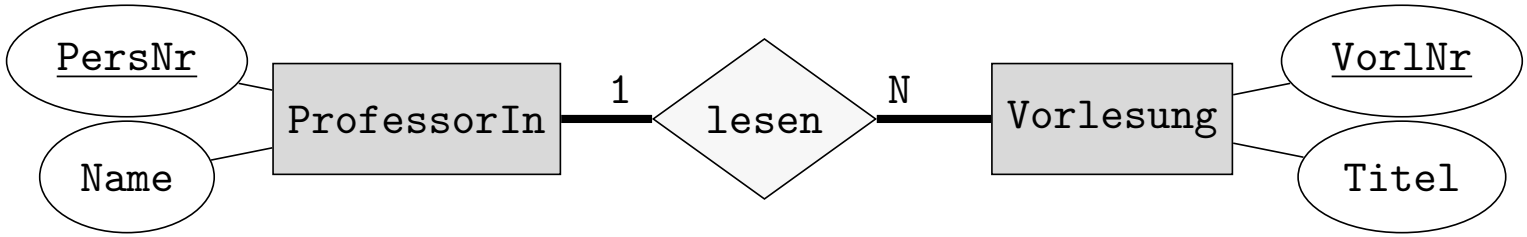
Schlüssel einer Relation, die aus einer N:M Beziehung entstanden ist, beinhaltet alle Fremdschlüsselattribute der an der Beziehung beteiligten Entities.

Beispiele

hören: (*MatrNr: integer, VorlNr: integer*) (N:M)

voraussetzen: (*Vorgänger: integer, Nachfolger: integer*) (N:M)

1:N Beziehung (lesen)



ProfessorIn	
PersNrNr	Name
2125	Sokrates
2126	Russel
2127	Kopernikus
2133	Popper
2134	Augustinus
2136	Curie

lesen	
PersNr	VorlNr
2137	5001
2125	5041
2125	5049
2125	4052
2126	5216
...	...

Vorlesung	
VorlNr	Titel
5001	Grundzüge
5041	Ethik
5049	Mäeutik
4052	Logik
5216	Bioethik
...	...

Schlüssel von 1:N Beziehungen

Schlüssel einer Relation, die aus einer 1:N Beziehung entstanden ist, beinhaltet jene Fremdschlüsselattribute, die von der "N" Seite der an der Beziehung beteiligten Entity stammen.

Beispiele

lesen: (*PersNr: integer*, <u>*VorlNr: integer*</u>) (1:N)

arbeitenFür: (<u>*AssiPersNr: integer*</u>, *ProfPersNr: integer*) (N:1)

prüfen: (<u>*MatrNr: integer, VorlNr: integer*</u>, *PersNr: integer*, Note: decimal) (N:M:1)

Verfeinerung von 1:N Beziehungen

- Wir hatten:

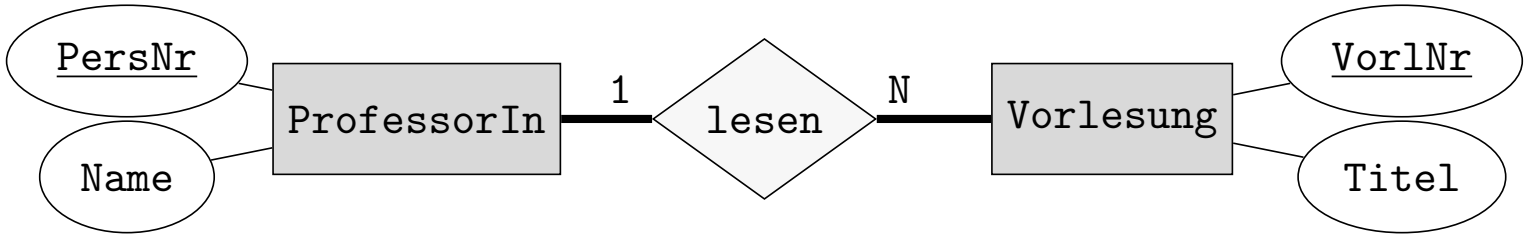
  ProfessorIn: (PersNr: integer, Name: string, Rang: string, Raum: integer)
  Vorlesung: (VorlNr: integer, Titel: string, SWS: integer)
  lesen: (*VorlNr: integer*, *PersNr: integer*)

- Verfeinerung durch Zusammenfassung von Relationen:

  ProfessorIn: (PersNr: integer, Name: string, Rang: string, Raum: integer)
  Vorlesung: (VorlNr: integer, Titel: string, SWS: integer, *PersNr: integer: Prof.PersNr*)

Nur Relationen mit gleichem Schlüssel zusammenfassen!

Ergebnis der richtigen Zusammenfassung



ProfessorIn			
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

Vorlesung			
VorlNr	Titel	SWS	PersNr
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5049	Mäeutik	2	2125
4052	Logik	4	2125
5216	Bioethik	2	2126
...	...	...	...

Falsches Zusammenfassen führt zu Anomalien

ProfessorIn

PersNr	Name	Rang	Raum	liest
2125	Sokrates	C4	226	5041
2125	Sokrates	C4	226	5049
2125	Sokrates	C4	226	4052
2126	Russel	C4	232	5216
...	...	...	...	...
2136	Curie	C4	36	???

Vorlesung

VorlNr	Titel	SWS
5001	Grundzüge	4
5041	Ethik	4
5049	Mäeutik	2
4052	Logik	4
5216	Bioethik	2
...	...	...

Problem Schlüssel: Relation ProfessorIn braucht neuen Schlüssel

Update-Anomalie: Sokrates zieht um

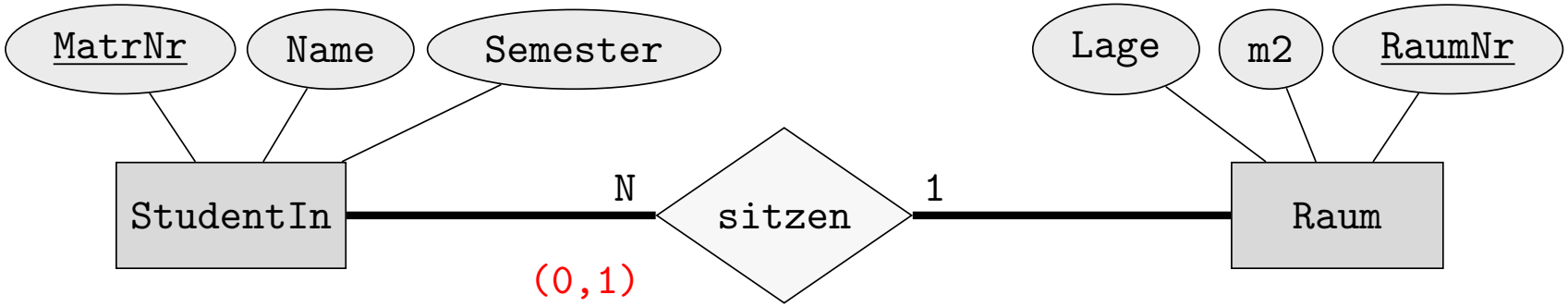
Einfüge-Anomalie: Curie ist neu und hält noch keine Vorlesung.
Was ist der Schlüssel?

Lösch-Anomalie: Vorlesung Ethik fällt weg

Nullwerte und deren Vermeidung

Beispiel

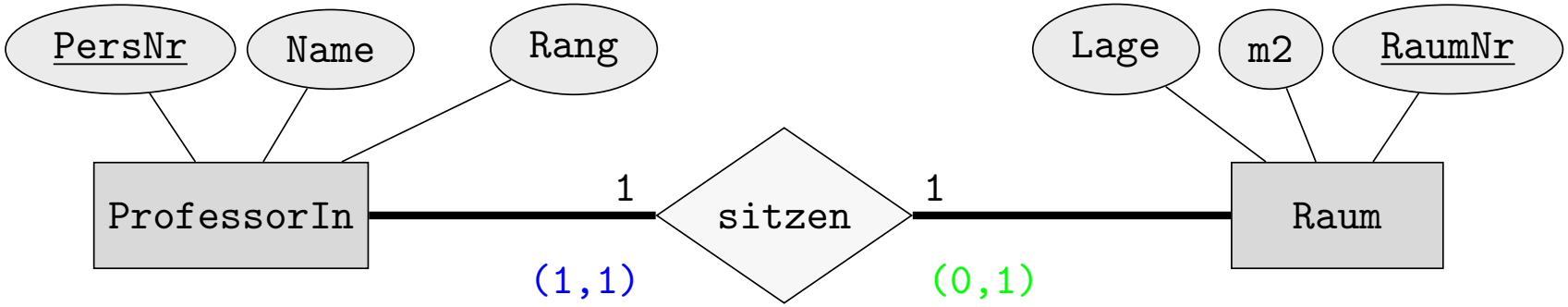
Studierende, die als StudienassistentInnen arbeiten, bekommen einen Arbeitsraum. Es gibt 25.000 Studierende und 200 StudienassistentInnen.



StudentIn: (MatrNr, Name, Semester)
sitzen: (*MatrNr: StudentIn, RaumNr: Raum*)
Raum: (RaumNr, m2, Lage)
~~StudentIn: (MatrNr, Name, Semester, *RaumNr: Raum*)~~

Hier nicht zusammenfassen um Nullwerte zu vermeiden

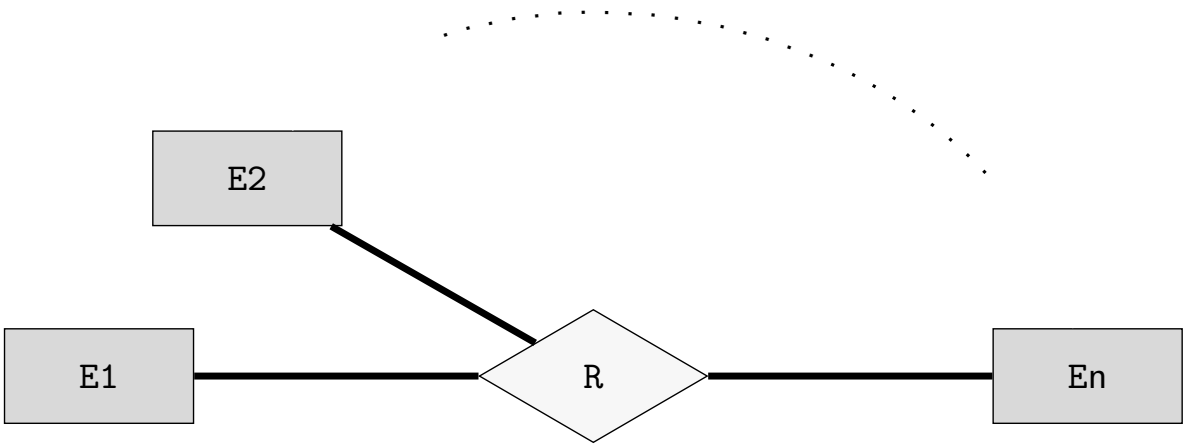
1:1 Beziehung (sitzen)



ProfessorIn: (PersNr, Name, Rang)
Raum: (RaumNr, m2, Lage)
sitzen: (*PersNr: ProfessorIn*, *RaumNr: Raum*) oder
(*PersNr: ProfessorIn* *RaumNr: Raum*)
ProfessorIn: (PersNr, Name, Rang, *RaumNr: Raum*)
Raum: (RaumNr, m2, Lage)
ProfessorIn: (PersNr, Name, Rang)
Raum: (RaumNr, m2, Lage, *PersNr: ProfessorIn*)

Achtung: Nullwertvermeidung

Verfeinerung bei (min,max)-Notation



$$R \subseteq E_1 \times E_2 \times \cdots \times E_n$$

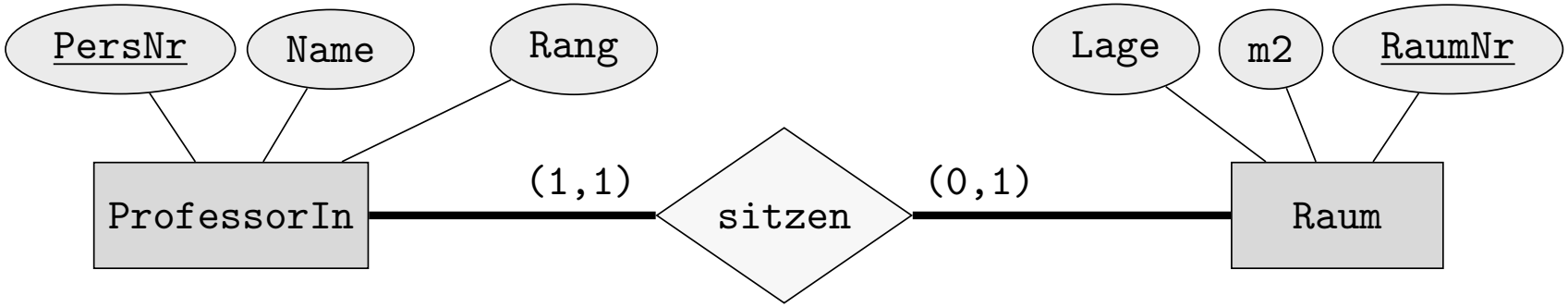
Übersetzung in ein relationales Schema:

$$R^s : (key(E_1), key(E_2), \ldots, key(E_n))$$

(min,max)-Notation: Für jede Entity e_i vom Typ E_i gilt:

$$\min_i \leq \#\text{Tupel der Art } (\ldots, e_i, \ldots) \in R \leq \max_i$$

Verfeinerung bei (min,max)-Notation



ProfessorIn: (PersNr, Name, Rang)

Raum: (RaumNr, m2, Lage)

sitzen: (*PersNr: ProfessorIn*, *RaumNr: Raum*)

ProfessorIn: (PersNr, Name, Rang, *RaumNr: Raum*)

Raum: (RaumNr, m2, Lage)

Verfeinerung bei (min,max)-Notation

Die (min,max)-Notation gibt an, wie oft eine Entity (des Typs E) in einer Ausprägung eines Beziehungstyps auftreten kann.

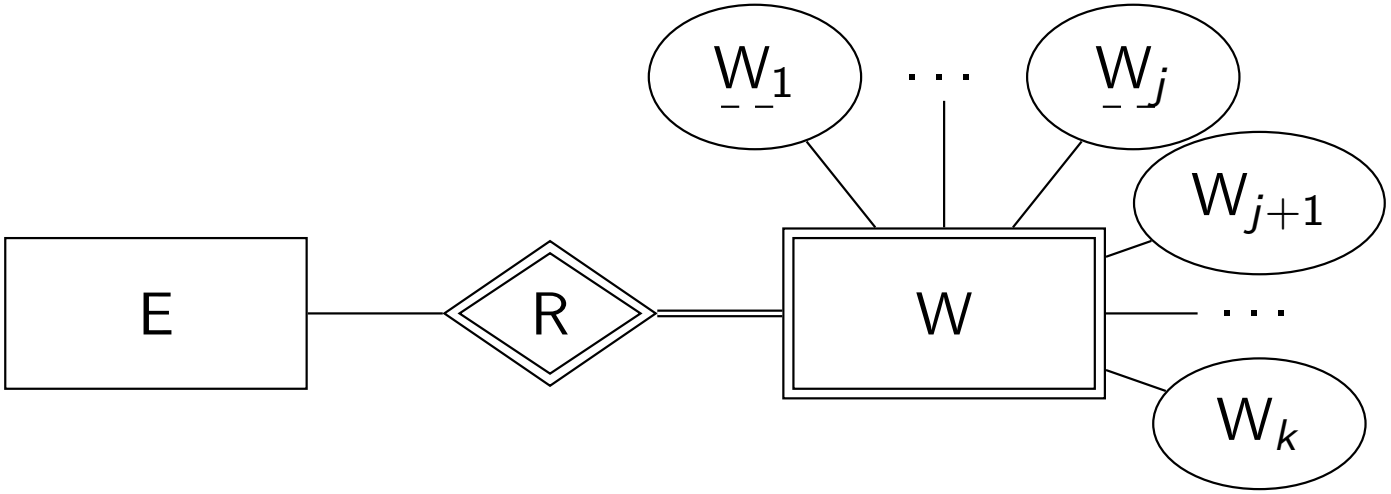
(0,1) Maximal Einmal $\Rightarrow$
Identifiziert Tupel der Beziehung eindeutig.

Schlüssel enthält die Fremdschlüsselattribute von E.
Zusammenfassen kann zu NULL-Werten führen.

(1,1) Genau Einmal $\Rightarrow$
Identifiziert Tupel der Beziehung eindeutig.
Jede Entity vom Typ E steht in einer Beziehung.

Schlüssel enthält die Fremdschlüsselattribute von E.
Fasse Beziehung mit der Relation E zusammen.

Relationale Darstellung schwacher Entitytypen



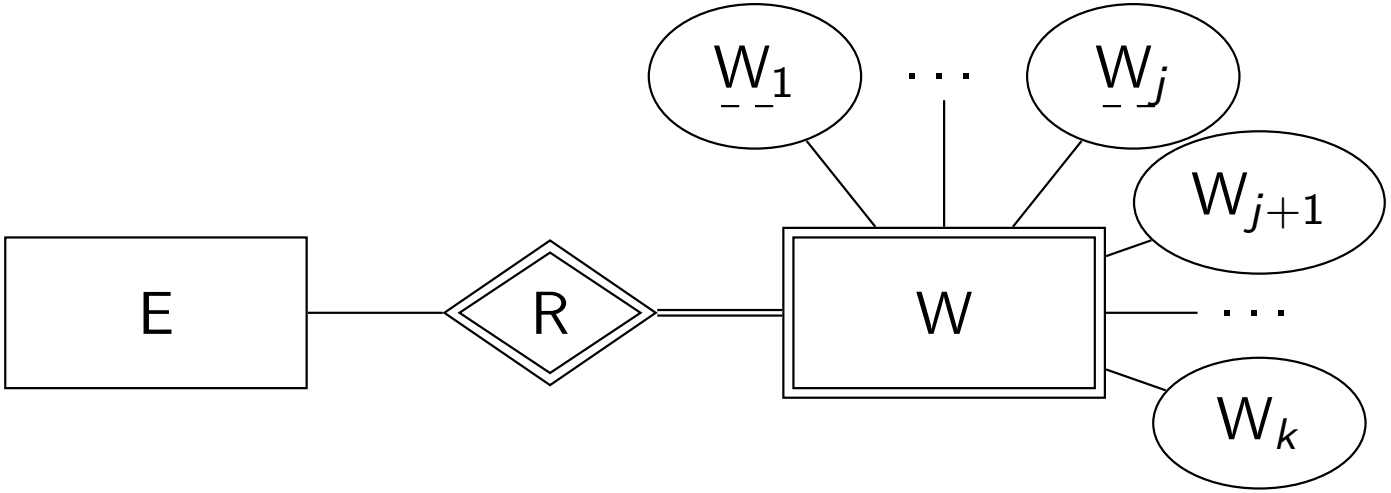
Intuitiv:

W (Schlüssel von *rel*(A),

Schwache Schlüssel von W,

Attribute von W)

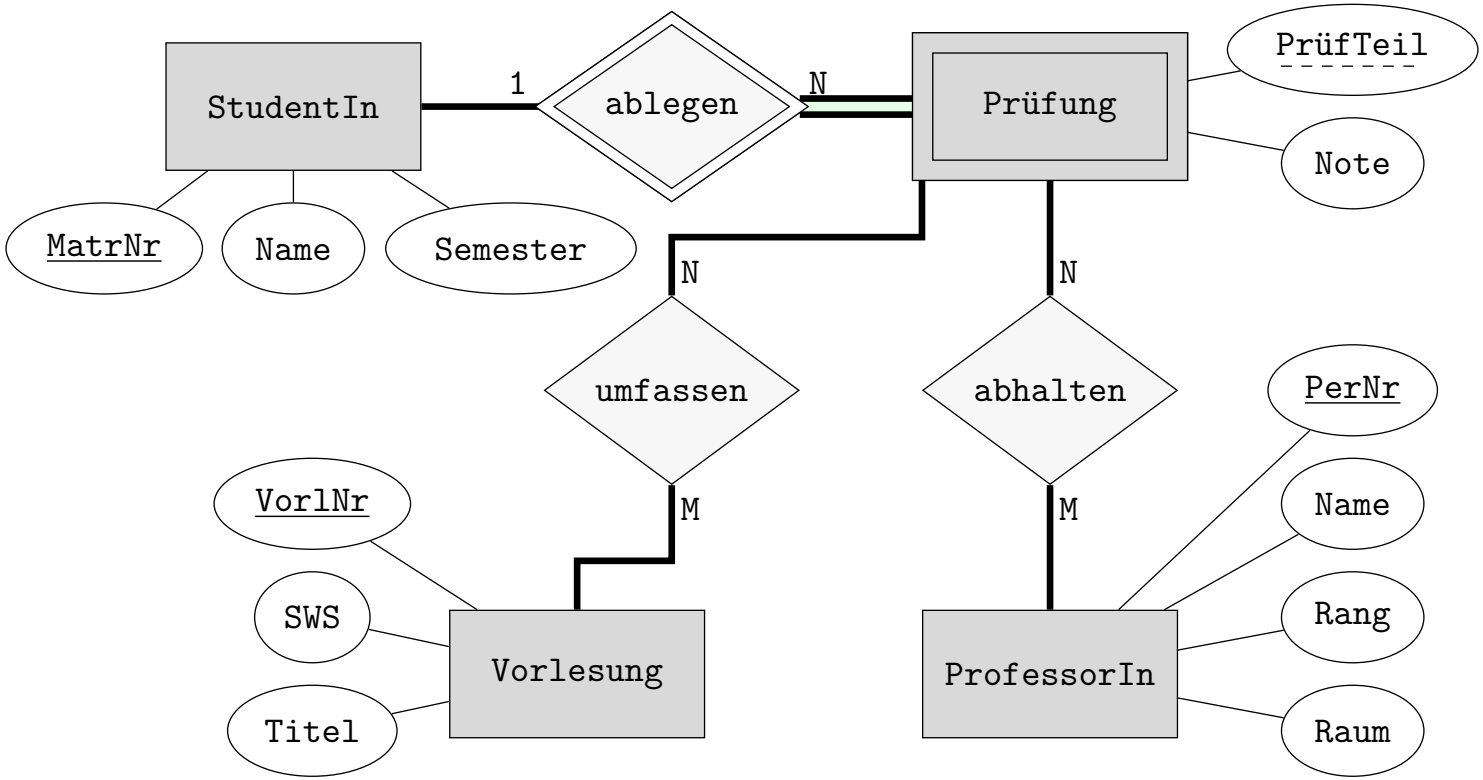
Relationale Darstellung schwacher Entitytypen



Annahme: E: ($\underline{A_1, \ldots, A_i}, A_{i+1}, \ldots, A_n$)

W: ($\underline{A_1 : E.A_1, \ldots, A_i : E.A_i}, W_1, \ldots, W_j, W_{j+1}, \ldots, W_k$)

Relationale Darstellung schwacher Entitytypen

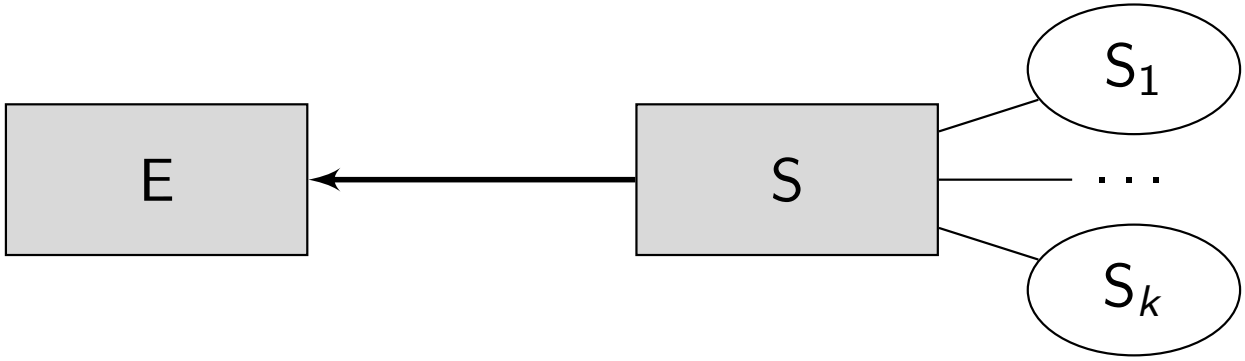


Prüfung: (*MatrNr: StudentIn.MatrNr*, PrüfTeil, Note)

umfassen: (*MatrNr: Prfg, PrüfTeil: Prfg, VorlNr: Vorl*)

abhalten: (*MatrNr: Prfg, PrüfTeil: Prfg, PersNr: Prof*)

Relationale Darstellung der Generalisierung



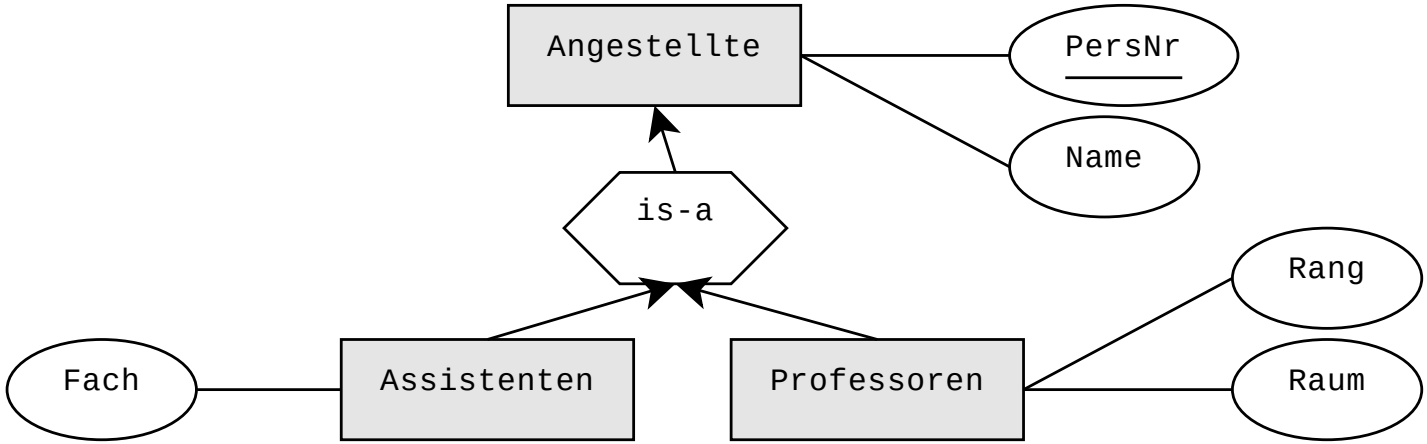
Intuitiv: "Erbe" den Schlüssel

Annahme: E: $(\underline{A_1, \ldots, A_i}, A_{i+1}, \ldots, A_k)$

S: $(\underline{A_1 : E.A_1, \ldots, A_i : E.A_i}, S_1, \ldots, S_k)$

Relationale Darstellung der Generalisierung

Relationale Darstellung der Generalisierung aus dem Uni Schema



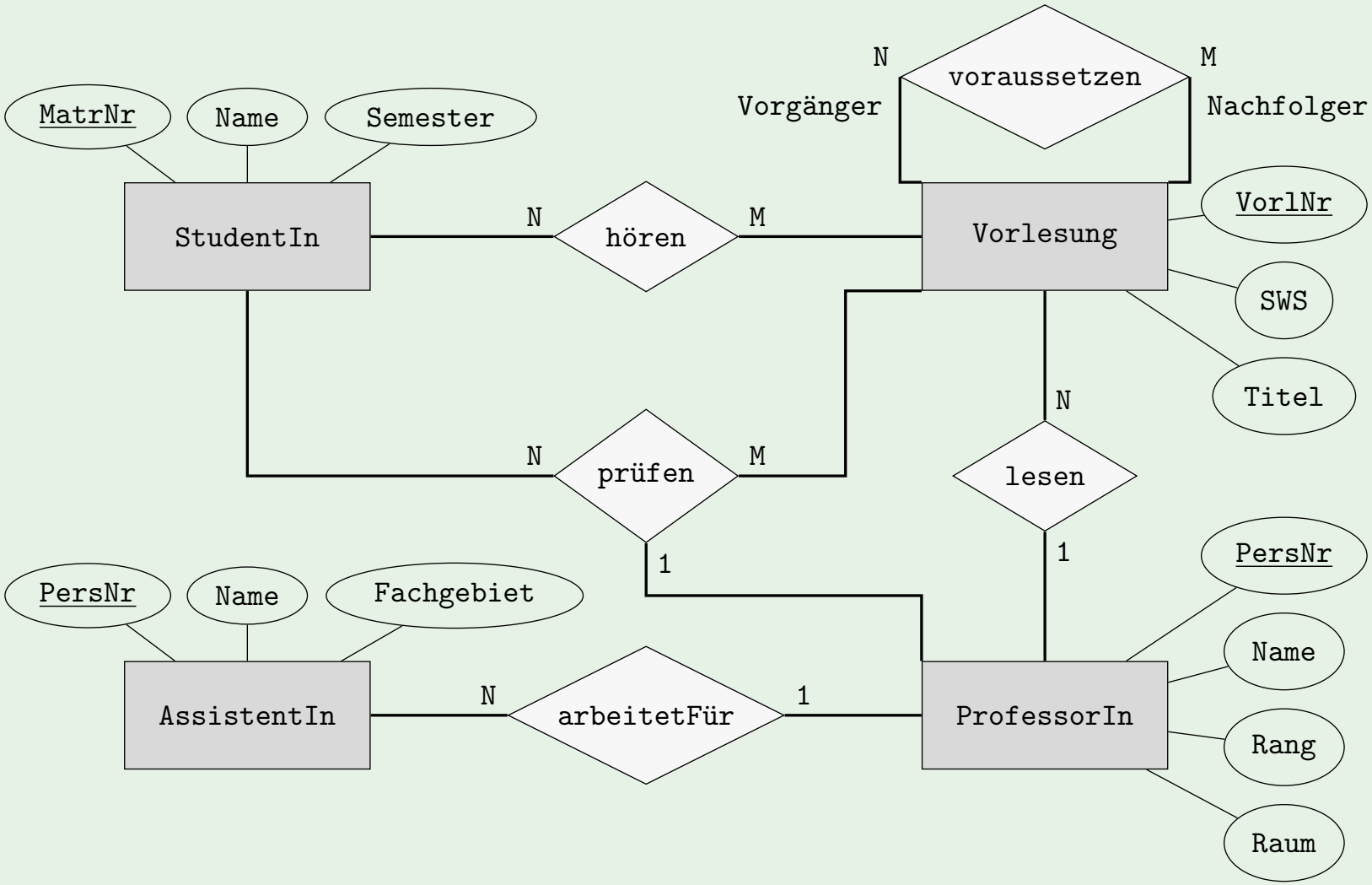
Angestellte: (PersNr, Name)

ProfessorIn: (*PersNr: Angestellte.PersNr*, Rang, Raum)

AssistentIn: (*PersNr: Angestellte.PersNr*, Fach)

Von EER zum relationalen Schema: Unischema

Beispiel (Unischema)



Relationale Universitätsdatenbank

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

ProfessorIn			
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

Relationale Universitätsdatenbank

Vorlesung

VorlNr	Titel	SWS	PersNr
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 drei Kritiken	4	2137

voraussetzen

VorgNr	NachfNr
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

Relationale Universitätsdatenbank

hören

MatrNr	VorlNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5001
28106	4052
28106	4630
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022

AssistentIn

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

prüfen

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

Lernziele

- Wie übersetzt man
  - Entitytypen
  - Beziehungstypen
  - schwache Entitytypen
  - Generalisierungsbeziehungen

  in das relationale Modell?
- Worauf ist bei der Übersetzung von Beziehungstypen zu achten?
  - Wie kann man die Schlüssel identifizieren?
  - Wie erhält man eine möglichst geringe Anzahl an Relationen?
  - Wie können NULL-Werte vermieden werden?
- Welche Arten von Anomalien können sich aus einem schlechten Schema ergeben?

Datenabfragesprachen

Datenabfragesprachen (Query Languages)

In dieser Einheit:

- Relationale Algebra
- Relationenkalkül

Beide Sprachen

- Bilden theoretische Grundlage für SQL
- Sind gleich ausdrucksstark
- Sind relational abgeschlossen

Relationale Algebra

Die relationale Algebra

CODD 1970: A relational model for large shared data banks.
Communications of the ACM, 13(6): 377-387

CODD 1972: Relational Completeness of Data Base Sublanguages.
In: Rustin, R., Hrsg.: Database Systems, 33-64
Prentice Hall, Englewood Cliffs, NY, USA

Prozedurale Abfragesprache:
Ausdruck beinhaltet implizit Abarbeitungsplan zur Ausführung der Abfrage

Mengenorientierte Sprache:
Operationen arbeiten auf Mengen von Tupeln

Relationale Algebra – Warum (lernen wir das)?

- Basis vieler (relationaler) Anfragesprachen
- Beschreibt mögliche Operationen auf Relationen
  (= Werkzeug und Denkweise)
- Verwendung (Auszug):
  - Erstellung und Optimierung von Anfrageplänen in DBMSs
  - Als Funktionen "prozeduraler" DB Interfaces
  - Kommunikation über Datenbankoperationen,
    Beschreibung möglicher Vorgehensweisen
  - Entwicklung von Algorithmen zur Anfragebeantwortung

Relationale Algebra – Warum (lernen wir das)?

- Erstellung und Optimierung von Anfrageplänen in DBMSs

Beispiel (Postgres Query Plan)

```
Hash Join
 Hash Cond: (w.ssn = e.essn)
 -> Hash Join
    Hash Cond: (w.pno = p.pnumber)
    -> Seq Scan on workson w
    -> Hash
      -> Bitmap Heap Scan on project p
         Recheck Cond: ((pname)::text = 'Aquarius'::text)
        -> Bitmap Index Scan on projectpnameidx
           Index Cond: ((pname)::text = 'Aquarius'::text)
 -> Hash
    -> Seq Scan on employee e
       Filter: ((bdate)::text > '1957-12-31'::text)
```

Relationale Algebra – Warum (lernen wir das)?

- Als Funktionen "prozeduraler" DB Interfaces

Beispiel (Berechnungen mit SPARK Data Frames)

```
mDF.join(pDF,pDF("made") === mDF("name"))
  .select(mDF("name"),pDF("name"))
  .except(mDF
    .join(pDF,pDF("made") === mDF("name"))
  .join(dDF,dDF("for") === pDF("id"))
  .select(mDF("name"),pDF("name")))
```

Die Operatoren der relationalen Algebra

Basisoperatoren

σ: Selektion

π: Projektion

$\cup$: Vereinigung

$-$: Mengendifferenz

$\times$: kartesisches Produkt (Kreuzprodukt)

ρ: Umbenennung

$\bowtie$: Join (Verbund)

$⟕, ⟖$ bzw. $⟗$: linker, rechter bzw. voller äußerer Join

$\rtimes$ bzw. $\ltimes$: linker bzw. rechter Semi-Join

$\cap$: Durchschnitt

$\div$: Division

Die Selektion: $\sigma_F(R)$

Beispiel

Finde alle Studierenden, die mehr als 10 Semester inskribiert sind.

$$\sigma_{\text{Semester}>10}(\text{StudentIn})$$

StudentIn

MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3

$\sigma_{\textbf{Semester}>10}(\textbf{StudentIn})$

MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12

Die Selektion: $\sigma_F(R)$

- Auswahl von Tupeln der Relation R mittels der Formel F
- F verwendet Vergleichsoperatoren ($=, \neq, \leq, \geq, >, <$), logische Operatoren ($\neg, \vee, \wedge$), Attributnamen von R und Konstanten

Definition ($\sigma_{\mathsf{F}}(\mathrm{R})$)

- Schema: $att(\sigma_{\mathsf{F}}(\mathrm{R})) = att(R)$
- Ausprägung: $\sigma_{\mathsf{F}}(\mathrm{R}) = \{t \in R \mid t \text{ erfüllt } F\}$

Auswertung von F für Tupel t: Ersetze alle Attributnamen in F durch den entsprechenden Wert in t

Die Projektion $\pi_{A_i}(R)$

Beispiel

Finde alle Rangbezeichnungen für ProfessorInnen.

$$\pi_{Rang}(ProfessorIn)$$

ProfessorIn			
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

$\pi_{Rang}(ProfessorIn)$
Rang
C3
C4

Die Projektion $\pi_{A_i}(R)$

- Auswahl einer Menge von Attributen A_i einer Relation E
- Achtung: Duplikate werden eliminiert

Definition ($\pi_{A_i}(R)$)

Sei A_i eine Teilmenge der Attribute von R

- Schema: $att(\pi_{A_i}(R)) = A_i$
- Ausprägung: $\{t' \mid \exists t \in R\colon t.A_i = t'\}$

Selektion und Projektion

StudentIn		
MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26830	Aristoxenos	8
27550	Schopenhauer	6
26120	Fichte	10
28106	Carnap	3

StudentIn		
MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10

StudentIn
MatrNr
24002
25403
26830
27550
26120
28106

Die Vereinigung $R \cup S$

Beispiel

Finde die Namen aller ProfessorInnen und AssistentInnen?

$$\pi_{Name}(ProfessorIn) \cup \pi_{Name}(AssistentIn)$$

$\pi_{\textbf{Name}}(\textbf{ProfessorIn})$
Name
Sokrates
...
Curie

$\pi_{\textbf{Name}}(\textbf{AssistentIn})$
Name
Platon
...
Spinoza

$\pi_{Name}(ProfessorIn) \cup \pi_{Name}(AssistentIn)$
Name
Sokrates
...
Kant
Platon
...
Spinoza

Die Vereinigung $R \cup S$

- Definiert auf zwei Relationen R, S mit gleichem Schema
- Gibt alle Tupel aus, die in R oder in S vorkommen

Definition ($R \cup S$)

- Vorbedingung: $att(R) = att(S)$
- Schema: $att(R \cup S) = att(R)$
- Ausprägung: $\{t \mid t \in R \text{ oder } t \in S\}$

Die Mengendifferenz $R - S$

Beispiel

Finde die (Matrikelnummer der) Studierenden, die noch keine Prüfung absolviert haben.

$$\pi_{MatrNr}(StudentIn) - \pi_{MatrNr}(prüfen)$$

$\pi_{\mathbf{MatrNr}}(\mathbf{StudentIn})$
MatrNr
24002
25403
26120
26830
27550
28106
29120
29555

$\pi_{MatrNr}(prüfen)$
MatrNr
28106
25403
27550

$\pi_{MatrNr}(StudentIn)$ – $\pi_{MatrNr}(prüfen)$
MatrNr
24002
26120
26830
29120
29555

Die Mengendifferenz $R - S$

- Definiert auf zwei Relationen R, S mit gleichem Schema
- Gibt alle Tupel aus, die in R aber nicht in S vorkommen

Definition ($R - S$)

- Vorbedingung: $att(R) = att(S)$
- Schema: $att(R - S) = att(R)$
- Ausprägung: $\{t \mid t \in R \text{ und } t \notin S\}$

Das Kartesische Produkt $R \times S$

Beispiel

Finde alle Paare von StudentIn und Einträgen in "hören"

StudentIn $\times$ hören

StudentIn $\times$ *hören*				
MatrNr	Name	Sem	hören.MatrNr	VorlNr
24002	Xenokrates	18	26120	5001
24002	Xenokrates	18	27550	5001
...	...	...	...	...
24002	Xenokrates	18	25403	5022
25403	Jonas	12	26120	5001
...	...	...	...	...
29555	Feuerbach	2	29555	5022
29555	Feuerbach	2	25403	5022

Das Kartesische Produkt $R \times S$

- Verknüpft jedes Tupel von R mit jedem Tupel von S
- Schema von $R \times S$ ist die Vereinigung der Attribute von R und S
- Ergebnisgröße: $|R \times S| = |R| * |S|$
- Oftmals "bessere" Operation ist der Join

Definition ($R \times S$)

Sei $att(R) = (A_1, \ldots, A_m)$ und $att(S) = (B_1, \ldots, B_n)$.

- Schema: $att(R \times S) = (A_1, \ldots, A_m, B_1, \ldots, B_n)$ (sicherstellen dass kein Attributname doppelt vorkommt)
- Ausprägung: $\{t \mid \exists t_1 \in R\colon t.[A_1, \ldots, A_m] = t_1$ und $\exists t_2 \in S\colon t.[B_1, \ldots, B_n] = t_2\}$

Die Umbenennung $\rho_X(R)$

Beispiel (Umbenennung von Attributen)

Finde die VorlNr aller Vorlesungen ohne Voraussetzungen

$$\rho_{\text{VorlNr} \leftarrow \text{NachfNr}}(\textit{voraussetzen})$$

Vorlesungen			
VorlNr	Titel	SWS	PersNr

voraussetzen	
VorgNr	NachfNr

$\rho_{\textbf{VorlNr} \leftarrow \textbf{NachfNr}}(\textbf{voraussetzen})$	
VorgNr	VorlNr

$$\pi_{\text{VorlNr}}(\text{Vorlesungen}) - \pi_{\text{VorlNr}}(\rho_{\text{VorlNr} \leftarrow \text{NachfNr}}(\text{voraussetzen}))$$

Die Umbenennung $\rho_X(R)$

Beispiel (Umbenennung von Relationen)

Finde alle Paare von Studierenden (MatrNr) welche (mindestens) eine Vorlesung gemeinsam hören.

hören	
MatrNr	VorlNr

hören	
MatrNr	VorlNr

$$\sigma_{S1.VorlNr=S2.VorlNr}\big(\rho_{S1}(\text{hören}) \times \rho_{S2}(\text{hören})\big)$$

S1	
MatrNr	VorlNr

S2	
MatrNr	VorlNr

ρ_{S1}(**hören**) $\times$ ρ_{S2}(**hören**)			
S1.MatrNr	S1.VorlNr	S2.MatrNr	S2.VorlNr

Die Umbenennung $\rho_X(R)$

- Die Umbenennung von Attributen $\rho_{A \leftarrow B}(R)$:

  $\rho_{A \leftarrow B}(R)$ benennt das Attribut B der Relation R neu mit A

- Die Umbenennung von Relationen $\rho_V(R)$

  Die Relation R bekommt den neuen Namen V

Die Primitiven Operatoren der Relationalen Algebra

$\sigma_F(R)$: Auswahl von Tupeln in R die F erfüllen

$\pi_{A_i}(R)$: Auswahl der Attribute A_i

ρ: Umbenennung von Attributen oder Relationen

$\cup$: Vereinigung

$-$: Mengendifferenz

$\times$ kartesisches Produkt (Kreuzprodukt)

Der natürliche Verbund (Join)

Beispiel (Eine sehr häufige Art der Abfrage)

Welche Studierende besuchen welche Vorlesungen?

StudentIn

MatrNr	Name	Sem
26120	Fichte	10
25403	Jonas	12
...	...	...

hören

MatrNr	VorlNr
26120	5001
27550	5001
...	...

Vorlesung

VorlNr	Titel	...
5001	Grundzüge	...
5041	Ethik	...
...	...	...

$$\sigma_{\text{StudentIn.MatrNr}=\text{hören.MatrNr}\wedge\text{Vorlesung.VorlNr}=\text{hören.VorlNr}}(\text{StudentIn} \times \text{hören} \times \text{Vorlesung})$$

Der natürliche Verbund (Join) $R \bowtie S$

Beispiel

Welche Studierende besuchen welche Vorlesungen?

$$\pi_{St.MNr,N,S,VNr}(\sigma_{\text{StudentIn.MatrNr}=\text{hören.MatrNr}}(\text{StudentIn} \times \text{hören}))$$

$$\text{StudentIn} \bowtie \text{hören}$$

StudentIn

MatrNr	Name	Sem
24002	Xenokrates	18
...	...	...

hören

MatrNr	VorlNr
26120	5001
...	...

StudentIn ⋈ hören

MatrNr	Name	Sem	VorlNr
26120	Fichte	10	5001
27550	Schopenhauer	6	5001
27550	Schopenhauer	6	4052
28106	Carnap	3	5041
28106	Carnap	3	5001
...	...	...	...

Der natürliche Verbund (Join) $R \bowtie S$

- Verknüpft zwei Relationen R, S.

  1. Bildet das kartesische Produkt der Relationen
  2. Selektiert jene Tupel, die auf den gleichnamigen Attributen denselben Wert annehmen
  3. Projiziert die doppelt vorkommenden Attribute weg
  4. (Degeneriert zum Kartesische Produkt falls die Relationen keine gleichnamigen Attribute besitzen)

$R \bowtie S$											
$att(R) \setminus att(S)$				$att(R) \cap att(S)$				$att(S) \setminus att(R)$			
A_1	A_2	...	A_m	B_1	B_2	...	B_k	C_1	C_2	...	C_n
...	...	...	...	...	...	...	...	...	...	...	...

Der natürliche Verbund (Join) $R \bowtie S$

Definition (Natürlicher Verbund)

Seien R, S mit folgenden Schemata gegeben: $R(A_1, \ldots, A_m, B_1, \ldots B_k)$ und $S(B_1, \ldots B_k, C_1, \ldots C_n)$. Der natürliche Verbund ist definiert als

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

$R \bowtie S$											
$att(R) \setminus att(S)$				$att(R) \cap att(S)$				$att(S) \setminus att(R)$			
A_1	A_2	$\ldots$	A_m	B_1	B_2	$\ldots$	B_k	C_1	C_2	$\ldots$	C_n
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$

Bsp: Der natürliche Join

Welche Studierende besuchen welche Vorlesungen?

StudentIn		
MatrNr	Name	Sem
24002	Xenokrates	18
25403	Jonas	12
...	...	...

hören	
MatrNr	VorlNr
26120	5001
27550	5001
...	...

Vorlesung			
VorlNr	Titel	SWS	PersNr
5001	Grundzüge	4	2137
5041	Ethik	4	2125
...	...	...	...

Bsp: Der natürliche Join

StudentIn ⋈ *hören* ⋈ *Vorlesung*

MatrNr	Name	Sem	VorlNr	Titel	SWS	PersNr
26120	Fichte	10	5001	Grundzüge	4	2137
27550	Schopenhauer	6	5001	Grundzüge	4	2137
27550	Schopenhauer	6	4052	Logik	4	2125
28106	Carnap	3	5041	Ethik	4	2125
28106	Carnap	3	5001	Grundzüge	4	2137
28106	Carnap	3	4052	Logik	4	2125
28106	Carnap	3	4630	Die drei Kritiken	4	2137
29120	Theophrastos	2	5001	Grundzüge	4	2137
29120	Theophrastos	2	5041	Ethik	4	2125
29120	Theophrastos	2	5049	Mäeutik	2	2125
29555	Feuerbach	2	5022	Glaube und Wissen	2	2134
25403	Jonas	12	5022	Glaube und Wissen	2	2134

Definition: Ausdrücke der relationalen Algebra

Definition (Relationale Algebra)

Die Basisausdrücke der relationalen Algebra sind:

- Relationen der Datenbank oder
- konstante Relationen

Seien R und S Ausdrücke der relationalen Algebra, so sind:

- $\sigma_F(R)$ und $\pi_{A_i}(R)$
- $R \cup S$, $R - S$, und $R \times S$
- $\rho_{A \leftarrow B}(R)$ und $\rho_V(R)$

ebenfalls gültige Ausdrücke der relationalen Algebra.

Ausdrücke der relationalen Algebra

Beispiel

$$\pi_{Name}(\sigma_{R.MN=St.MN}(\rho_R(\pi_{MN}(StudentIn) - \pi_{MN}(hören)) \times StudentIn)) \cup \pi_{Name}(ProfessorIn)$$

ProfessorIn

StudentIn

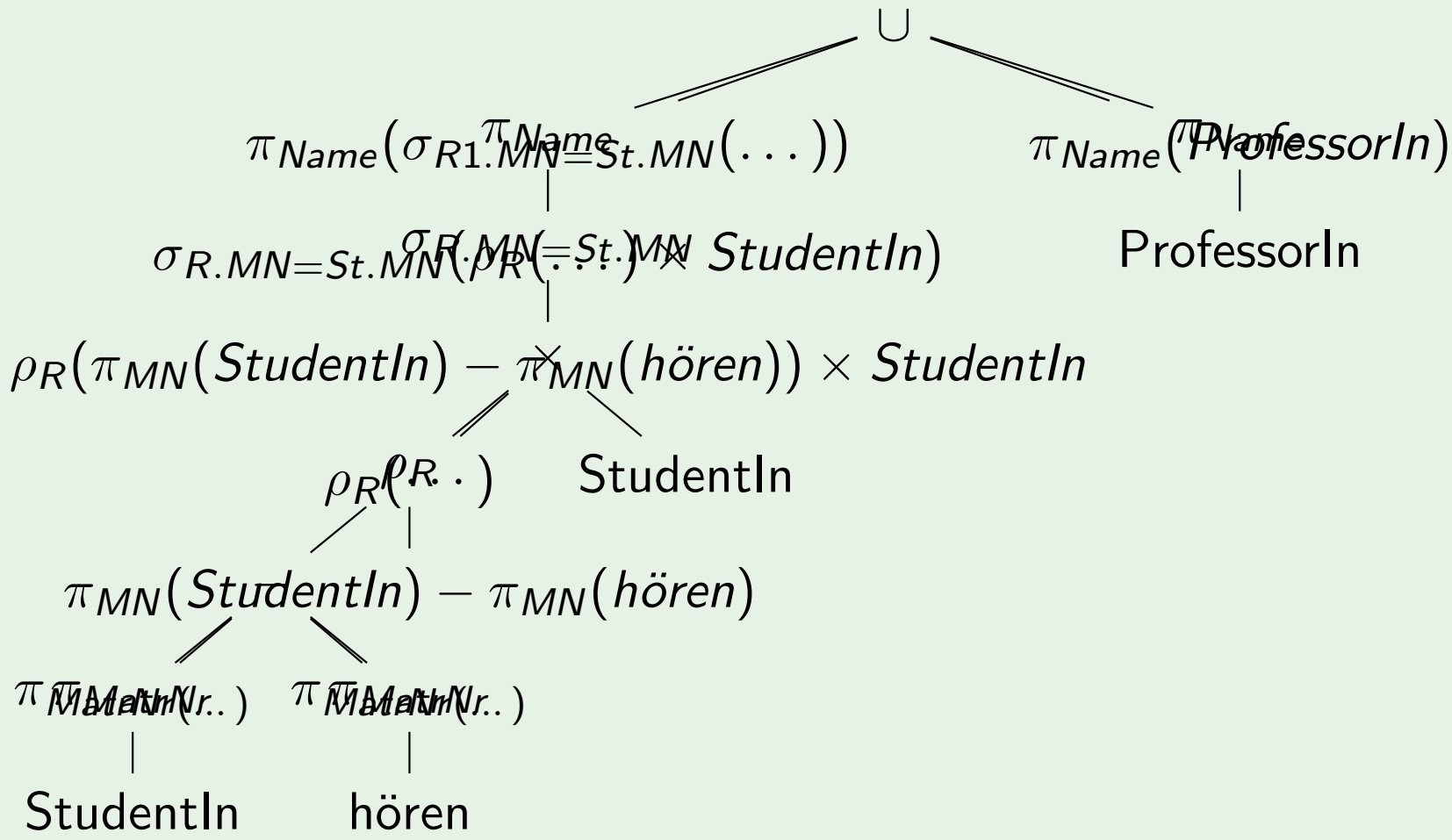
StudentIn

hören

Ausdrücke der relationalen Algebra

Beispiel

$$\pi_{Name}(\sigma_{R.MN=St.MN}(\rho_R(\pi_{MN}(StudentIn) - \pi_{MN}(hören)) \times StudentIn)) \cup \pi_{Name}(ProfessorIn)$$



Beispiel: Auswerten eines Ausdrucks der RA

Schema:

$$R(A, B, C),\ S(B, D, E),\ T(A, C, D)$$

Anfrage:

$$\pi_{ACD}(\sigma_{E<5 \wedge C \neq 2}(R \bowtie S)) - T$$

R		
A	B	C
1	2	3
7	1	4
2	4	3
3	4	7
1	1	1
7	4	2

S		
B	D	E
8	3	1
4	9	9
4	4	1
1	7	7
3	7	7

T		
A	C	D
1	3	3
3	7	4
7	4	9
7	2	7

A	C	D
2	3	4
3	7	4
3	4	7
2	4	3
3	4	7
3	4	7
1	1	1
7	4	2
7	4	2

D	E
4	1
4	1
9	9
4	1
9	9
4	1
7	7
9	9
4	1

Der allgemeine Verbund (Join) $R \bowtie_\theta S$

- Verknüpft zwei Relationen R, S, auch wenn sie keine gleichnamigen Attribute haben

  aufgrund einer logischen Bedingung θ.

Definition

Seien R und S mit folgenden Schemata gegeben:
$R(A_1, \ldots, A_n)$ und $S(B_1, \ldots, B_m)$.
Sei θ ein Prädikat über den Attributen $A_1, \ldots, A_n, B_1, \ldots, B_m$.
Der allgemeine Verbund ist definiert als

$$R \bowtie_\theta S = \sigma_\theta(R \times S)$$

Der allgemeine Verbund (Join) $R \bowtie_\theta S$

- Verknüpft zwei Relationen R, S, auch wenn sie keine gleichnamigen Attribute haben

  aufgrund einer logischen Bedingung θ.

Beispiele (θ)

$$\begin{aligned}
\theta &= Sem > 10 \wedge SWS = 4 \vee PersNr = 2134 \\
\theta &= StudentIn.MatrNr = hören.MatrNr \wedge \\
&\quad hören.VorlNr = Vorlesungen.VorlNr \\
\theta &= Vorlesungen.VorlNr = voraussetzen.VorgNr
\end{aligned}$$

Andere Join Arten

natürlicher Join (Wh): nur jene Tupel bleiben im Ergebnis, die einen "Join-Partner" gefunden haben.

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

$=$

$L \bowtie R$				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1

voller äußerer Join: alle Tupel bleiben erhalten

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

$=$

$L ⟗ R$				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_2	NULL	NULL
NULL	NULL	c_3	d_2	e_2

Andere Join Arten: Linker/Rechter äußerer Join

linker äußerer Join: Tupel der linken Relation bleiben erhalten

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

$=$

$L ⟕ R$				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_2	NULL	NULL

rechter äußerer Join: Tupel der rechten Relation bleiben erhalten

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

$=$

$L ⟖ R$				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
NULL	NULL	c_3	d_2	e_2

Andere Join Arten: Semi-Join

Semi-Join von L mit R (bzw. R mit L): alle Tupel der Relation L (bzw. R), die einen Join eingehen können, werden ausgewählt.

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

$=$

$L \ltimes R$		
A	B	C
a_1	b_1	c_1

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

$=$

$L \rtimes R$		
C	D	E
c_1	d_1	e_1

Der Mengendurchschnitt $R \cap S$

- Definiert auf zwei Relationen R, S mit gleichem Schema
- Gibt alle Zeilen aus, die in R und in S vorkommen
- Kann mittels Mengendifferenz ausgedrückt werden:

$$R \cap S = R - (R - S)$$

Beispiel

Finde die PersNr jener C4 ProfessorInnen, die eine Vorlesung halten.

$\pi_{PersNr}(Vorlesungen) \cap \pi_{PersNr}\sigma_{Rang=C4}(ProfessorIn)$
PersNr
2137
2125
2126
2137

Die relationale Division $R \div S$

Beispiel

Welche Studierenden haben alle 4-stündigen Vorlesungen gehört?

hören

MatrNr	VorlNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5001
28106	4052
28106	4630
29120	5001
29120	5041
29120	5049

$\div$

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

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

VorlNr
5001
5041
4052
4630

$=$

$R \div S$

MatrNr
28106

Die relationale Division $R \div S$

- Definiert auf zwei Relationen R, S
- Schema von S muss Teilmenge von Schema von R sein
- Ergebnisschema: Attribute von R ohne jenen von S
- Ergebnis enthält alle Tupel von $\pi_{att(R) \setminus att(S)}(R)$ die mit jedem Tupel in S ein Tupel in R bilden
- Gegenstück zum kartesischen Produkt:

$$T = U \times V \Rightarrow T \div U = V \text{ und } T \div V = U$$

- Kann ausgedrückt werden als

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

(mit $\mathcal{R} = att(R)$ und $\mathcal{S} = att(S)$)

Die relationale Division $R \div S$

Definition

Seien R, S Relationen, mit $\mathcal{S} \subseteq \mathcal{R}$. Tupel $t \in R \div S$ wenn es für jedes Tupel $s \in S$ ein Tupel $r \in R$ gibt, mit:

$$\begin{aligned} r.\mathcal{S} &= s \\ r.(\mathcal{R} - \mathcal{S}) &= t \end{aligned}$$

Weitere Beispiele

Beispiel

Finde die Vorlesungsnummern der Vorlesungen, die indirekte Vorgänger 2. Stufe der VO 5216 sind
(= Vorgänger der Vorgänger von 5216)

$$\rho_{V_1}(\text{voraussetzen}) \quad \rho_{V2}(\text{voraussetzen})$$

voraussetzen	
VorgNr	NachfNr

V_1	
VorgNr	NachfNr

voraussetzen	
VorgNr	NachfNr

V_2	
VorgNr	NachfNr

$$\pi_{V_1.VorgNr}\Big(\sigma_{V_1.NachfNr=V_2.VorgNr \wedge V_2.NachfNr=5216}$$
$$\big(\rho_{V_1}(voraussetzen) \times \rho_{V2}(voraussetzen)\big)\Big)$$

Lernziele

- Wann ist eine Abfragesprache relational abgeschlossen?
- Wodurch sind Operatoren der relationalen Algebra definiert?
- Welche Operatoren der relationalen Algebra gibt es?
- Was sind gültige Ausdrücke der relationalen Algebra?

Übersicht

Datenabfragesprachen

- Relationale Algebra
- Relationenkalkül
- Ausdruckskraft von Abfragesprachen

Relationenkalkül

Der Relationenkalkül

Deklarative Abfragesprache: spezifiziert, welche Daten gefordert sind, nicht, wie sie erhalten werden

Mengenorientierte Sprache: Operationen arbeiten auf Mengen von Tupeln

Die Anfragen im Relationenkalkül sind von der Form:

$$\{t \mid P(t)\}$$

wobei $P(t)$ eine Formel ist.
Es gibt zwei unterschiedliche, aber gleichmächtige Ausprägungen:

- der relationale Tupelkalkül
- der relationale Domänenkalkül

Tupelkalkül

Der relationale Tupelkalkül

Beispiele

ProfessorIn(PersNr, Name, Rang, Raum)

- Finde alle C4-ProfessorInen:

$$\{p \mid p \in \mathit{ProfessorIn} \wedge p.\mathit{Rang} = \text{‘C4’}\}$$

- Finde die Namen aller C4-ProfessorInnen:

$$\{[p.\mathit{Name}] \mid p \in \mathit{ProfessorIn} \wedge p.\mathit{Rang} = \text{‘C4’}\}$$

Der relationale Tupelkalkül

Beispiele

ProfessorIn(PersNr, Name, Rang, Raum)
AssistentIn(PersNr, Name, GebDatum, Boss)

- Finde Paare von ProfessorInnen und die ihnen zugeordneten AssistentInnen:

$$\begin{aligned} \{[p.Name, a.PersNr] \mid p &\in \text{ProfessorIn} \wedge \\ a &\in \text{AssistentIn} \wedge \\ & p.PersNr = a.Boss\} \end{aligned}$$

Der relationale Tupelkalkül

Beispiel

$$\{[p.Name, a.PersNr] \mid p \in \text{Profs} \wedge a \in \text{Assis} \wedge p.\text{PersNr} = a.\text{Boss}\}$$

ProfessorIn

	PersNr	Name	...	...
$p =$	2125	Sokrates	...	...
$p =$	2126	Russel	...	...
	2127	Kopernikus	...	...

AssistentIn

PersNr	...	...	Boss	
3002	...	...	2125	$= a$
3003	...	...	2125	$= a$
3004	...	...	2126	$= a$

Ergebnis

Name	PersNr
Sokrates	3002
Sokrates	3003
Russel	3004

Der relationale Tupelkalkül

Beispiele

StudentIn(MatrNr, Name, Semester)
hören(MatrNr, VorlNr)
Vorlesungen(VorlNr, Titel, SWS, gelesenVon)

- Studierende die mindestens eine Vorlesung besuchen:

$$\{s \mid s \in StudentIn \wedge \exists h \in hören(s.MatrNr = h.MatrNr)\}$$

- MatrNr der Studierenden die alle Vorlesungen besuchen:

$$\begin{aligned}\{[s.MatrNr] \mid s \in\ & StudentIn \wedge \forall v \in Vorlesungen(\\ & \exists h \in hören(h.VorlNr = v.VorlNr \\ & \wedge h.MatrNr = s.MatrNr))\}\end{aligned}$$

Der relationale Tupelkalkül

Beispiel (Abfragen im Tupelkalkül)

StudentIn

	MatrNr	...
s	24002	...
s	25403	...
s	26120	...
s	26830	...

Vorlesung

	VorlNr	Titel	...
v	5041	Ethik	...
v	5049	Mäeutik	...
v	4052	Logik	...
v	5216	Bioethik	...

hören

MatrNr	VorlNr	
26120	5041	
26830	5041	
29120	5049	
24002	4052	h
26120	4052	
25403	5216	
26120	5216	

$$\{ \quad [s.MatrNr] \mid s \in StudentIn \wedge \forall v \in Vorlesungen(\exists h \in hören(h.VorlNr = v.VorlNr \wedge h.MatrNr = s.MatrNr))\}$$

Der relationale Tupelkalkül

Definition (Syntax)

Die Anfragen des relationalen Tupelkalküls sind von der Form:

$$\{t \mid P(t)\}$$

t ... Tupelvariable (oder Tupelkonstruktor); $P(t)$ ... Formel.
Formeln werden aus Atomen zusammengebaut (siehe Definition).

Definition (Semantik)

Ein Tupel t ist im Ergebnis, wenn es die Formel $P(t)$ erfüllt.
Die Variable t ist eine freie Variable der Formel $P(t)$, ist also nicht durch einen Quantor ($\forall, \exists$) gebunden.

Der relationale Tupelkalkül

Definition (Syntax: Atome und Formeln)

Atome:

$t \in R$: t Tupelvariable; R Relationenname

$s.A\phi t.B$: s und t Tupelvariablen; A und B Attributnamen; ϕ ein Vergleichsperator $(=, \neq, <, \leq, >, \geq)$

$s.A\phi c$: wie oben, aber c Konstante

Formeln:

Atome sind Formeln

P Formel, so sind auch $\neg P$ und (P) Formeln

P_1, P_2 Formeln, so sind auch $P_1 \wedge P_2, P_1 \vee P_2$ und $P_1 \rightarrow P_2$ Formeln

$P(t)$ Formel mit einer freien Variable t, so sind auch $\forall t \in R(P(t))$ und $\exists t \in R(P(t))$ Formeln

Der relationale Tupelkalkül

Beispiele

StudentIn(MatrNr, Name, Semester)
hören(MatrNr, VorlNr)
Vorlesungen(VorlNr, Titel, SWS, gelesenVon)
ProfessorIn(PersNr, Name, Rang, Raum)

- Studierende die mindestens eine Vorlesung von Curie besuchen:

$$\begin{aligned}\{s \mid s \quad &\in \quad \mathit{StudentIn} \wedge \exists h \in \mathit{hören}(s.\mathit{MatrNr} = h.\mathit{MatrNr} \\ &\wedge \quad \exists v \in \mathit{Vorlesungen}(h.\mathit{VorlNr} = v.\mathit{VorlNr} \\ &\wedge \quad \exists p \in \mathit{ProfessorIn}(p.\mathit{PersNr} = v.\mathit{gelesenVon} \\ &\wedge \quad p.\mathit{Name} = \text{‘Curie’})))\}\end{aligned}$$

Der relationale Tupelkalkül

Beispiele

StudentIn(MatrNr, Name, Semester)
Vorlesungen(VorlNr, Titel, SWS, gelesenVon)
hören(MatrNr, VorlNr)

- Studierende die alle vierstündigen Vorlesungen besuchten:

$$\begin{aligned}\{s \mid s \quad \in \quad & StudentIn \wedge \forall v \in Vorlesungen(v.SWS \neq 4 \vee \\ & \exists h \in hören(h.VorlNr = v.VorlNr \\ & \wedge h.MatrNr = s.MatrNr))\}\end{aligned}$$

Der relationale Tupelkalkül

Beispiele

StudentIn(MatrNr, Name, Semester)
AssistentIn(PersNr, Name, GebDatum, Boss)
ProfessorIn(PersNr, Name, Rang, Raum)

- PersonalNummern aller ProfessorInnen und AssistentInnen:

$$\{[p.PersNr] \mid p \in ProfessorIn \vee p \in AssistentIn\}$$

- Namen die sowohl AssistentInnen als auch Studierende tragen:

$$\{[p.Name] \mid p \in AssistentIn \wedge \exists s \in StudentIn(p.Name = s.Name)\}$$

Der relationale Tupelkalkül

Beispiele

AssistentIn(PersNr, Name, GebDatum, Boss)
ProfessorIn(PersNr, Name, Rang, Raum)

- Alle AssistentenInnen die keine ProfessorInnen sind:

$$\{a \mid a \in Asssistenten \wedge \neg\exists p \in ProfessorIn(p.PersNr = a.PersNr)\}$$

$$\{a \mid a \in Asssistenten \wedge \forall p \in ProfessorIn(p.PersNr \neq a.PersNr\}$$

Domänenkalkül

Der relationale Domänenkalkül

Beispiele

ProfessorIn(PersNr, Name, Rang, Raum)

- Finde alle C4-ProfessorInnen:

$$\{[p, n, r, o] \mid ([p, n, r, o] \in \mathit{ProfessorIn} \wedge r = \text{‘C4’})\}$$

- Finde die Namen aller C4-ProfessorInnen:

$$\{[n] \mid \exists p, o, r([p, n, r, o] \in \mathit{ProfessorIn} \wedge r = \text{‘C4’})\}$$

Der relationale Domänenkalkül

Beispiele

ProfessorIn(PersNr, Name, Rang, Raum)
AssistentIn(PersNr, Name, GebDatum, Boss)

- Finde Paare von ProfessorInnen und die ihnen zugeordneten AssistentInnen:

$$\{[n, a] \mid \exists\ p, r, o([p, n, r, o] \in \mathit{ProfessorIn} \wedge \exists\ m, f([a, m, f, p] \in \mathit{AssistentIn}))\}$$

Der relationale Domänenkalkül

Beispiele

StudentIn(MatrNr, Name, Semester)
hören(MatrNr, VorlNr)
Vorlesungen(VorlNr, Titel, SWS, gelesenVon)

- Studierende die mindestens eine Vorlesung besuchen:

$$\{[m, n] \mid \exists s([m, n, s] \in \mathit{StudentIn} \wedge \exists v([m, v] \in \mathit{hören}))\}$$

- MatrNr der Studierenden die alle Vorlesungen besuchen:

$$\begin{aligned}\{[m] \quad \mid \quad & \exists n, s([m, n, s] \in \mathit{StudentIn} \wedge \forall v, t, s, p \\ & ([v, t, s, p] \in \mathit{Vorlesungen} \rightarrow [m, v] \in \mathit{hören})\}\end{aligned}$$

Der relationale Domänenkalkül

Definition (Syntax)

Die Anfragen des relationalen Domänenkalküls sind von der Form:

$$\{[v_1, v_2, \ldots, v_n] \mid P(v_1, v_2, \ldots, v_n)\}$$

$v_1, v_2, \ldots, v_n$ ... Domänenvariablen, die einen Attributwert repräsentieren; $P(v_1, v_2, \ldots, v_n)$ eine Formel. Formeln werden aus Atomen zusammengebaut (siehe Definition).

Definition (Semantik)

Ein Tupel $[v_1, v_2, \ldots, v_n]$ ist im Ergebnis, wenn es die Formel $P(v_1, v_2, \ldots, v_n)$ erfüllt. Die Variablen $v_1, v_2, \ldots, v_n$ sind freie Variablen der Formel $P(v_1, v_2, \ldots, v_n)$ – also nicht durch einen Quantor ($\forall, \exists$) gebunden.

Der relationale Domänenkalkül

Definition (Syntax: Atome und Formeln)

Atome:

$[v_1, \ldots, v_n] \in R$: $v_1, v_2, \ldots, v_n$ Domänenvariablen; R ein n-stelliger Relationenname

$x \phi y$: x, y Domänenvariablen; ϕ ein Vergleichsperator $(=, \neq, <, \leq, >, \geq)$

$x \phi c$: wie oben, aber c Konstante

Formeln:

Atome sind Formeln

P Formel, so sind auch $\neg P$ und (P) Formeln

P_1, P_2 Formeln, so auch $P_1 \wedge P_2, P_1 \vee P_2$ und $P_1 \Rightarrow P_2$

$P(v)$ Formel mit freien Variablen v, so sind auch $\forall v(P(v))$ und $\exists v(P(v))$ Formeln

Der relationale Domänenkalkül

Beispiele

StudentIn(MatrNr, Name, Semester)
hören(MatrNr, VorlNr)
Vorlesungen(VorlNr, Titel, SWS, gelesenVon)
ProfessorIn(PersNr, Name, Rang, Raum)

- Name und MatrNr von Studierenden die mindestens eine Vorlesung von Curie besuchen:

$$\begin{aligned}\{[m,n] \mid\ & \exists s([m,n,s] \in \mathit{StudentIn} \wedge \\ & \exists v([m,v] \in \mathit{hören} \wedge \\ & \exists t,d,p([v,t,d,p] \in \mathit{ProfessorIn} \wedge \\ & \exists a,r,u([p,a,r,u] \in \mathit{ProfessorIn} \wedge \\ & a = \text{‘Curie’}))))\}\end{aligned}$$

Der relationale Domänenkalkül

Beispiele

StudentIn(MatrNr, Name, Semester)
hören(MatrNr, VorlNr)
Vorlesungen(VorlNr, Titel, SWS, gelesenVon)
ProfessorIn(PersNr, Name, Rang, Raum)

- Name und MatrNr von Studierenden die mindestens eine Vorlesung von Curie besuchen:

$$\begin{aligned}\{[m,n] \mid\ & \exists s,v,t,d,p,r,u([m,n,s] \in \mathit{StudentIn} \wedge \\ & [m,v] \in \mathit{hören} \wedge [v,t,d,p] \in \mathit{ProfessorIn} \wedge \\ & [p,\text{`}\mathit{Curie}',r,u] \in \mathit{ProfessorIn}))\}\end{aligned}$$

Der relationale Domänenkalkül

Beispiele

StudentIn(MatrNr, Name, Semester)
hören(MatrNr, VorlNr)
Vorlesungen(VorlNr, Titel, SWS, gelesenVon)

- Name und MatrNr von Studierenden die alle vierstündigen Vorlesungen besuchen:

$$\begin{aligned}\{[m,n] \quad | \quad & \exists s([m,n,s] \in \mathit{StudentIn} \wedge \\ & \forall v,t,s,p(([v,t,s,p] \in \mathit{Vorlesungen} \wedge s = 4) \\ & \rightarrow [m,v] \in \mathit{hören})\}\end{aligned}$$

Der relationale Domänenkalkül

Beispiele

StudentIn(MatrNr, Name, Semester)
AssistentIn(PersNr, Name, GebDatum, Boss)
ProfessorIn(PersNr, Name, Rang, Raum)

- PersonalNummern aller ProfessorInnen und AssistentInnen:

$$\{[p] \mid \quad (\exists n, g, b([p, n, g, b] \in AssistentIn) \vee (\exists r, a([p, n, r, a] \in ProfessorIn)\}$$

- Namen die sowohl AssistentInnen als auch Studierende tragen:

$$\{[n] \mid \exists p, g, b, m, s(\quad [p, n, g, b] \in AssistentIn \wedge [m, n, s] \in StudentIn)\}$$

Der relationale Domänenkalkül

Beispiele

AssistentIn(PersNr, Name, GebDatum, Boss)
ProfessorIn(PersNr, Name, Rang, Raum)

- Alle AssistentInnen die keine ProfessorInnen sind:

$$\{[p] \mid \exists n, g, b \quad ([p, n, g, b] \in \mathit{Asssistenten} \wedge \neg\exists n, r, a([p, n, r, a] \in \mathit{ProfessorIn}))\}$$

$$\{[p] \mid \exists n, g, b \quad ([p, n, g, b] \in \mathit{Asssistenten} \wedge \forall n, r, a(\neg([p, n, r, a] \in \mathit{ProfessorIn})))\}$$

Vergleich: Tupel- und Domänenkalkül

Beispiel

Gesucht sind Studierende, die mindestens eine Vorlesung besuchen:

StudentIn(MatrNr, Name, Semester)
hören(MatrNr, VorlNr)

- Tupelkalkül:

$$\{s \mid s \in StudentIn \wedge \exists h \in hören(s.MatrNr = h.MatrNr)\}$$

- Domänenkalkül:

$$\{[m, n, s] \mid [m, n, s] \in StudentIn \wedge \exists v([m, v] \in hören)\}$$

Vergleich: Tupel- und Domänenkalkül

Beispiel

Gesucht sind die Matrikelnummern jener Studierenden, die auf alle abgelegten Prüfungen eine 2 hatten:
StudentIn(MatrNr, Name, Semester)
prüfen(MatrNr, VorlNr, Note)

- Tupelkalkül:

$$\{[s.MatrNr] \mid s \in StudentIn \wedge \forall p \in prüfen(s.MatrNr = p.MatrNr \rightarrow p.Note = 2)\}$$

- Domänenkalkül:

$$\{[mnr] \mid \exists nam, sem([mnr, nam, sem] \in StudentIn) \wedge \forall vnr, no([mnr, vnr, no] \in prüfen \rightarrow no = 2)\}$$

Vergleich: Tupel- und Domänenkalkül

Beispiel (Semantik Tupelkalkül)

StudentIn

	MatrNr	Name	Sem
$s =$	26120	Fichte	10
$s =$	29555	Feuerbach	2
$s =$	28106	Carnap	3
$s =$	27550	Schopenhauer	6

prüfen

MatrNr	VorlNr	Note
28106	5041	2 ✓✗
27550	5001	2 ✓✗
28106	5001	2 ✓✗
26120	5001	2 ✓✗
27550	4052	1 ✓✗
25403	5001	1 ✓✗

Erg.

MatrNr
1 ?
26120 ?
29555 ?
28106 ?
27550 ?

$$\{[s.MatrNr] \mid s \in StudentIn \wedge \forall p \in prüfen(s.MatrNr = p.MatrNr \rightarrow p.Note = 2)\}$$

$s = (1,1,1,1)?$ $s \in StudentIn?$ $\Rightarrow$ ✗

$s \in StudentIn$ ✓ $\forall p \in prüfen \ldots ?$ ✓ $\Rightarrow$ ✓

$s \in StudentIn$ ✓ $\forall p \in prüfen \ldots ?$ ✓ $\Rightarrow$ ✓

Vergleich: Tupel- und Domänenkalkül

Beispiel (Semantik Domänenkalkül)

StudentIn

MatrNr	Name	Sem
26120	Fichte	10
29555	Feuerbach	2
28106	Carnap	3
27550	Schopenhauer	6

prüfen

MatrNr	VorlNr	Note
28106	5041	2 ✓✗
27550	5001	2 ✓✗
28106	5001	2 ✓✗
26120	5001	2 ✓✗
27550	4052	1 ✓✗
25403	5001	1 ✓✗

Erg.

MatrNr
1 ?
26120 ?
29555 ?
28106 ?
27550 ?

$$\{[mnr] \mid \exists nam, sem([mnr, nam, sem] \in StudentIn) \wedge \forall vnr, no([mnr, vnr, no] \in prüfen \rightarrow no = 2)\}$$

$\exists nam, sem\colon [mnr, nam, sem] \in StudentIn?$ ✗ $\Rightarrow$ ✗

$\exists nam, sem\colon [mnr, nam, sem] \in StudentIn?$ ✓

$vnr = 5001$, $no = 2$ ✓ $\Rightarrow$ ✓

Sichere Ausdrücke
und
Ausdruckskraft der Abfragesprachen

Sichere Ausdrücke im Relationenkalkül

"Eine Anfrage im Relationenkalkül ist sicher, wenn Ihre Antwort nur von jenen Werten abhängt welche in der Datenbank oder der Anfrage vorkommen, jedoch nicht von dem angenommenen Wertebereich."

Beispiel (Keine sichere Anfrage)

$R(A)$

$$\{t \mid \neg(t \in R)\} \qquad \{[a] \mid \neg([a] \in R)\}$$

Sichere Ausdrücke im Tupelkalkül

Anfragen können unter Umständen eine unendliche Ergebnismenge spezifizieren:

Beispiel

$$\{n \mid \neg(n \in ProfessorIn)\}$$

gibt alle jene Tupel aus, die nicht in der Tabelle ProfessorIn vorkommen. Davon können wir uns unendlich viele vorstellen.

Sichere Ausdrücke im Tupelkalkül

Definition (Domäne eines Ausdrucks)

Die Domäne einer Formel enthält

- alle in der Formel vorkommenden Konstanten und
- alle Attributwerte von Relationen, die in der Formel referenziert werden.

Definition (Sichere Ausdrücke)

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

Sichere Ausdrücke im Domänenkalkül

Anfragen können unter Umständen wieder eine unendliche Ergebnismenge spezifizieren:

Beispiel

$$\{[p, n, r, o] \mid \neg([p, n, r, o] \in ProfessorIn)\}$$

gibt wieder alle jene Tupel aus, die nicht in der Tabelle ProfessorIn vorkommen. Davon können wir uns unendlich viele vorstellen.

Sichere Ausdrücke im Domänenkalkül

Definition (Sichere Ausdrücke)

Ein Ausdruck

$$\{[x_1, x_2, \ldots, x_n] \mid P(x_1, x_2, \ldots, x_n)\}$$

ist sicher, falls:

- Für jedes Tupel $[c_1, c_2, \ldots, c_n]$ im Ergebnis: c_i $(1 \leq i \leq n)$ ist in der Domäne von P enthalten
- Für jede Teilformel $\exists x P_1(x)$: Wenn Konstante c das Prädikat $P(c)$ erfüllt, dann ist c in der Domäne von P_1 enthalten
- Für jede Teilformel $\forall x P_1(x)$: dann und nur dann erfüllt, wenn $P_1(x)$ für alle Werte der Domäne von P_1 erfüllt

(2. und 3. im Tupelkalkül automatisch erfüllt!)

Ausdruckskraft der Abfragesprachen

Die drei Sprachen

- relationale Algebra,
- relationaler Tupelkalkül, eingeschränkt auf sichere Ausdrücke und
- relationaler Domänenkalkül, eingeschränkt auf sichere Ausdrücke

sind gleich mächtig

Wichtig: SQL ist ebenfalls gleich mächtig (mit bestimmten Einschränkungen)

Zum Abschluss . . .

Beispiel: Sport (aus: Formale Modellierung)

$$
\begin{aligned}
\mathcal{P} &= \{\mathit{Mensch}/1, \mathit{Sportart}/1, \mathit{Anstrengend}/1, \mathit{Betreibt}/2\} \\
\mathcal{F} &= \{\mathit{handball}/0, \mathit{laufen}/0\} \\
\mathcal{U} &= \{\text{Maria, Martina, Max, Moritz, Basketball, Fußball,} \\
&\quad \text{Handball, Schach, Singen, Slackline, Turnen, Laufen}\} \\
I(\mathit{Mensch}) &= \{\text{Martina, Max, Moritz }\} \\
I(\mathit{Sportart}) &= \{\text{Basketball, Fußball, Schach, Turnen }\} \\
I(\mathit{Anstrengend}) &= \{\text{Laufen, Basketball, Handball, Singen }\} \\
I(\mathit{Betreibt}) &= \{(\text{Maria, Handball}), (\text{Maria, Singen}), (\text{Maria, Fußball}), \\
&\quad (\text{Martina, Laufen}), (\text{Martina, Slackline}), \\
&\quad (\text{Martina, Basketball}), (\text{Max, Schach}), (\text{Max, Laufen}), \\
&\quad (\text{Max, Turnen}), (\text{Moritz, Laufen}), (\text{Moritz, Handball}) \} \\
I(\mathit{handball}) &= \text{Handball} \\
I(\mathit{laufen}) &= \text{laufen}
\end{aligned}
$$

Beispiel

$$\exists x(\mathit{Sportart}(x) \wedge \forall y(\mathit{Mensch}(y) \supset \neg \mathit{Betreibt}(y, x)))$$

Beispiel: Sport (aus: Formale Modellierung)

$$
\begin{aligned}
\mathcal{P} &= \{\mathit{Mensch}/1, \mathit{Sportart}/1, \mathit{Anstrengend}/1, \mathit{Betreibt}/2\} \\
I(\mathit{Mensch}) &= \{\text{Martina, Max, Moritz }\} \\
I(\mathit{Sportart}) &= \{\text{Basketball, Fußball, Schach, Turnen }\} \\
I(\mathit{Anstrengend}) &= \{\text{Laufen, Basketball, Handball, Singen }\} \\
I(\mathit{Betreibt}) &= \{(\text{Maria, Handball}), (\text{Maria, Singen}), (\text{Maria, Fußball}), \\
&\quad (\text{Martina, Laufen}), (\text{Martina, Slackline}), \\
&\quad (\text{Martina, Basketball}), (\text{Max, Schach}), (\text{Max, Laufen}), \\
&\quad (\text{Max, Turnen}), (\text{Moritz, Laufen}), (\text{Moritz, Handball})\ \}
\end{aligned}
$$

Beispiel

- $\exists x(\mathit{Sportart}(x) \wedge \forall y(\mathit{Mensch}(y) \supset \neg \mathit{Betreibt}(y,x)))$
- $\{(x) \mid (\mathit{Sportart}(x) \wedge \forall y(\mathit{Mensch}(y) \supset \neg \mathit{Betreibt}(y,x)))\}$
- $\{[x] \mid$
  $([x] \in \mathit{Sportart} \wedge \forall y([y] \in \mathit{Mensch} \rightarrow \neg([y,x] \in \mathit{Betreibt})))\}$

Lernziele

- Was sind die Unterschiede zwischen der relationalen Algebra und dem Relationenkalkül?
- Wann ist eine Sprache relational vollständig?
- Was sind die Unterschiede zwischen Tupel- und Domänenkalkül?
- Wie sehen Ausdrücke im Tupel- bzw. Domänenkalkül aus?
- Was sind sichere Ausdrücke im Relationenkalkül?