4. Relationenalgebra

- Einleitung
- Selektion und Projektion
- Mengenoperatoren
  - Vereinigung, Durchschnitt, Differenz
  - kartesisches Produkt
- Verbundoperationen (Join)
  - Theta-Join
  - natürlicher Verbund
  - Semi-Join
  - äußerer Verbund
- Division
- Beispielanfragen

Sprachen für das Relationenmodell

- Datenmodell = Datenobjekte + Operatoren
- im RM wird vereinheitlichte Sprache angestrebt für:
  - Anfragen (Queries) im 'Stand-Alone'-Modus
  - Datenmanipulation und Anfragen eingebettet in eine Wirtssprache
  - Datendefinition
  - Zugriffs- und Integritätskontrolle
  - Unterstützung verschiedener Benutzerklassen: Anwendungsprogrammierer, DBA, gelegentliche Benutzer
- verschiedene Grundtypen von Sprachen
  - formale Ansätze: Relationenalgebra und Relationenkalkül
  - abbildungsorientierte Sprachen (z. B. SQL)
  - graphik-orientierte Sprachen (z. B. Query-by-Example)

Relationenalgebra

- *Algebra*: ein System, das aus einer nichtleeren Menge und einer Familie von Operationen besteht
  - Relationen sind Mengen
  - Operationen auf Relationen arbeiten auf einer oder mehreren Relationen als Eingabe und erzeugen eine Relation als Ausgabe (Abgeschlossenheitseigenschaft)
    => mengenorientierte Operationen
- Operationen:

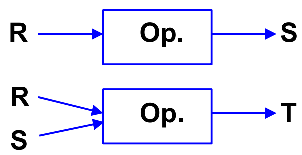
***Klassische Mengenoperationen*:**

- Vereinigung
- Differenz
- kartesisches Produkt
- Durchschnitt (ableitbar)

***Relationenoperationen*:**

- Restriktion (Selektion)
- Projektion
- Verbund (Join) (ableitbar)
- Division (ableitbar)

- 1-stellige und 2-stellige Operationen



Selektion (Restriktion)

- Auswahl von Zeilen einer Relation über Prädikate, abgekürzt σ_P

$$\boxed{\sigma_P(R) = \{ t \mid t \in R \wedge P(t)\}}$$

P = log. Formel (ohne Quantoren !) zusammengestellt aus:
- Operanden: Attributnamen oder Konstanten
- Vergleichsoperatoren $\theta \in \{<, =, >, \leq, \neq, \geq\}$
- logische Operatoren: $\vee, \wedge, \neg$

- Beispiele:

$\sigma_{\text{GEHALT} < \text{PROVISION}}$ (PERS)

$\sigma_{\text{BERUF='Programmierer'} \wedge \text{ALTER} < 50}$ (PERS)

- Eigenschaften
  - grad $(\sigma_P(R))$ = grad (R)
  - card $(\sigma_P(R))$ <= card (R)

Projektion

- Auswahl der Spalten (Attribute) $A_1, A_2, \ldots, A_k$ aus einer Relation R (Grad n >= k)

$$\pi_{A1,\, A2,\, \ldots,\, Ak}(R) = \{ p \mid \exists\, t \in R : p = \langle t[A_1], \ldots, t[A_k] \rangle \}$$

- Beispiel:

$$\pi_{NAME,\ GEHALT}(PERS)$$

- Eigenschaften:
  - wichtig: Duplikate werden entfernt ! (Mengeneigenschaft)
    z.B. für π_{ANR} (PERS)
  - grad $(\pi_A(R))$ <= grad (R)
  - card $(\pi_A(R))$ <= card (R)

Relationenalgebra: Beispiel-DB

ABT

ANR	ANAME	ORT
K51	Planung	Leipzig
K53	Einkauf	Frankfurt
K55	Vertrieb	Frankfurt

PERS

PNR	NAME	ALTER	GEHALT	ANR	MNR
406	Abel	47	50700	K55	123
123	Schulz	32	43500	K51	-
829	Müller	36	40200	K53	406
574	Schmid	28	36000	K55	123

- Finde alle Angestellten aus Abteilung K55, die mehr als 40.000 verdienen

  $\sigma_{ANR='K55' \wedge GEHALT > 40000}(PERS)$

- Finde alle Abteilungsorte

  $\pi_{ORT}(ABT)$

ORT
Leipzig
Frankfurt

- Finde den Abteilungsnamen von Abteilung K53

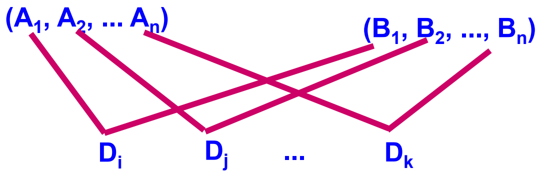
  $\pi_{ANAME}(\sigma_{ANR='K53'}(ABT))$

Klassische Mengenoperationen

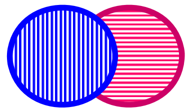
- Voraussetzung: *Vereinigungsverträglichkeit* der beteiligten Relationen:

  gleicher Grad - gleiche Bereiche:

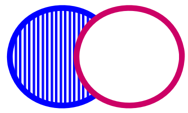
  => $W(A_i) = W(B_i)$: $i = 1, n$



- Vereinigung: $R \cup S = \{ t \mid t \in R \vee t \in S \}$
  - card ($R \cup S$) <= card (R) + card (S)

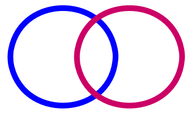


- Differenz: $R - S = \{ t \mid t \in R \wedge t \notin S \}$
  - card (R - S) <= card (R)



- Durchschnitt:

  $R \cap S = R - (R - S) = \{ t \mid t \in R \wedge t \in S \}$
  - card ($R \cap S$) <= min (card (R), card (S))



- Beispielanfrage: Welche Abteilungen (ANR) haben keine Mitarbeiter?

(Erweitertes) Kartesisches Produkt

R (Grad r) und S (Grad s) beliebig

$$\boxed{R \times S = \{ k \mid \exists x \in R, y \in S : k = x \mid y \}}$$

Beachte: $k = x \mid y = < x_1, \ldots, x_r, y_1, \ldots, y_s >$

<u>nicht</u> $<<x_1, \ldots, x_r>, <y_1, \ldots, y_s>>$ wie übliches kart. Produkt

- grad ($R \times S$) = grad (R) + grad (S)
- card ($R \times S$) = card (R) $*$ card (S)

Beispiel

R

A	B	C
a	γ	1
d	α	2
b	β	3

S

D	E	F
b	γ	3
d	α	2

$R \times S$

A	B	C	D	E	F
a	γ	1	b	γ	3
a	γ	1	d	α	2
d	α	2	b	γ	3
d	α	2	d	α	2
b	β	3	b	γ	3
b	β	3	d	α	2

Allgemeiner Verbund (Theta-Join)

- grob:
  - kartesisches Produkt zwischen zwei Relationen R (Grad r) und S (Grad s).
  - eingeschränkt durch Θ -Bedingungen zwischen Attribut A von R und Attribut B von S.
- Θ-Verbund zwischen R und S:

$$\boxed{R \underset{A\,\Theta\,B}{\bowtie} S = \sigma_{A\,\Theta\,B}(R \times S)}$$

mit arithm. Vergleichsoperator $\Theta \in \{<, =, >, \leq, \neq, \geq\}$

  - grad $(R \underset{A\Theta B}{\bowtie} S)$ = grad (R x S) = grad(R) + grad (S)
  - card $(R \underset{A\Theta B}{\bowtie} S)$ <= card (R x S)

- für häufigen Fall des ***Gleichverbunds (Equijoin)*** gilt Θ = '=' :

Natürlicher Verbund (Natural Join)

- grob: Gleichverbund über <u>alle</u> gleichnamigen Attribute und Projektion über die verschiedenen Attribute
- natürlicher Verbund zwischen R und S:

gegeben: $R(A_1, A_2, \ldots, A_{r-j+1}, \ldots, A_r)$, $S(B_1, B_2, \ldots, B_j, \ldots, B_s)$

o.B.d.A.:(sonst. Umsortierung: $B_1 = A_{r-j+1}$, $B_2 = A_{r-j+2}$ ... $B_j = A_r$

$$\boxed{R \bowtie S = \pi_{A_1,\ldots,A_r,B_{j+1},\ldots,B_s}\,\sigma_{(R.A_{r-j+1} = S.B_1) \wedge \ldots \wedge (R.A_r = S.B_j)}(R \times S)}$$

$\bowtie$ Zeichen für Natural Join ⇒ Θ = '='

  - Join-Attribute sind durch Übereinstimmungsbedingung gegeben
  - grad $(R \bowtie S)$ = r+s-j

R

A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$\bowtie$

S

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

Join-Beispiel

ABT

<u>ANR</u>	ANAME	ORT
K51	Planung	Leipzig
K53	Einkauf	Frankfurt
K55	Vertrieb	Frankfurt

PERS

<u>PNR</u>	NAME	ALTER	GEHALT	ANR	MNR
406	Abel	47	50700	K55	123
123	Schulz	32	43500	K51	-
829	Müller	36	40200	K53	406
574	Schmid	28	36000	K55	123

- Finde alle Angestellten (PNR, ALTER, ANAME), die in einer Abteilung in Frankfurt arbeiten und älter als 30 sind

$$\pi_{PNR,ALTER,ANAME}\ (\ \sigma_{ORT='Frankfurt'\ \wedge\ ALTER>30}\ (PERS \bowtie ABT))$$

<u>PNR</u>	ALTER	ANAME
406	47	Vertrieb
829	36	Einkauf

Semi-Join

- Ergebnisbeschränkung des Gleichverbundes auf eine der beiden Eingaberelationen

$$R \ltimes S = \pi_{\text{R-Attribute}}\ (S \bowtie R)$$ *linker Semijoin*

$$R \rtimes S = \pi_{\text{S-Attribute}}\ (S \bowtie R)$$ *rechter Semijoin*

R

A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$\ltimes$

S

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

R

A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$\rtimes$

S

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

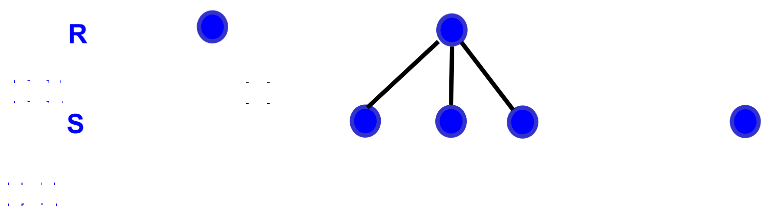
Äußerer Verbund (Outer Join)

- Ziel: verlustfreier Verbund soll erzwungen werden
  - Gleichverbund zwischen R und S ist *verlustfrei*, wenn alle Tupel von R und S am Verbund teilnehmen. Die inverse Operation Projektion erzeugt dann wieder R und S (**lossless join**).

$$R \bowtie S \text{ verlustfrei} \Leftrightarrow \pi_{\text{R-Attribute}}(R \bowtie S) = R \wedge \pi_{\text{S-Attribute}}(R \bowtie S) = S \Leftrightarrow$$

$$R \ltimes S = R \;\wedge\; R \rtimes S = S$$

- bisher: $R \bowtie S$ liefert nur „vollständige Objekte“
  - es sollen aber auch Teilobjekte als Ergebnis geliefert werden (z. B. komplexe Objekte)
  - Trick: Einfügen künstlicher Verbundpartner, um verlustfreien Verbund zu erreichen



Outer Join (2)

- Definition: seien A die Verbundattribute, $\{\equiv\}$ der undefinierte Wert und

$$\mathbf{R'} := R \cup ((\pi_A(S) - \pi_A(R)) \times \{\equiv\} \times \ldots \times \{\equiv\})$$
$$\mathbf{S'} := S \cup ((\pi_A(R) - \pi_A(S)) \times \{\equiv\} \times \ldots \times \{\equiv\})$$

äußerer Gleichverbund

$$R \underset{R.A = S.A}{⟗} S := R' \underset{R'.A = S'.A}{\bowtie} S'$$

*äußerer **natürlicher** Verbund*

$$R ⟗ S := R' \bowtie S'$$

- linker und rechter äußerer Gleichverbund
  - nur die linke bzw. rechte Eingaberelation bleibt verlustfrei (Einfügen künstlicher Verbundpartner in rechter bzw. linker Eingaberelation)

$$R \underset{R.A=S.A}{⟕} S := R \underset{R.A=S'.A}{\bowtie} S'$$

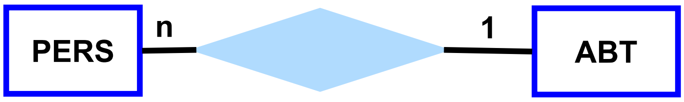
***linker** äußerer Gleichverbund*

$$R \underset{R.A=S.A}{⟖} S := R' \underset{R'.A=S.A}{\bowtie} S$$

***rechter** äußerer Gleichverbund*

- verallgemeinerbar auf 2 (oder mehr) Joins, z.B. $R ⟗ S ⟗ T$
  - selbst isolierte Tupel können zu einem vollständigen Pfad expandiert werden

Outer Join - Beispiel



PERS

PNR	ANR ...
P1	A1
P2	A1
P3	A2
P4	-
P5	-

ABT

ANR	ANAME ...
A1	A
A2	B
A3	C

PERS ⋈ ABT

PNR	ANR	ANAME ...
P1	A1	A
P2	A1	A
P3	A2	B

PERS ⟕ ABT

PNR	ANR	ANAME ...
P1	A1	A
P2	A1	A
P3	A2	B

PNR	ANR	ANAME ...
P1	A1	A
P2	A1	A
P3	A2	B

PERS ⟗ ABT

PNR	ANR	ANAME ...
P1	A1	A
P2	A1	A
P3	A2	B

PERS ⟖ ABT

Division

- Beantwortung von Fragen, bei denen eine „ganze Relation“ zur Qualifikation herangezogen wird
- Simulation des Allquantors => ein Tupel aus R steht mit allen Tupeln aus S in einer bestimmten Beziehung
- **Definition**

Voraussetzung: S-Attribute $\subset$ R-Attribute

sei R vom Grad r und S vom Grad s, $r > s$

t sei (r-s)-Tupel, u sei s-Tupel;

dann gilt: $R \div S = \{ t \mid \forall u \in S : t\,u \in R \}$

grad $(R \div S) = r - s$

card $(R \div S) <=$ card (R)

Division (2)

- Beispiel

LPT

LNR	PNR	TNR
L1	P1	T1
L1	P2	T1
L2	P1	T1
L2	P1	T2
L2	P2	T1

÷

PT

PNR	TNR
P1	T1
P1	T2
P2	T1

=

LNR
L2

  – welche Lieferanten beliefern alle Projekte?

$$\pi_{LNR,PNR}\ (LPT) \div \pi_{PNR}\ (PT)$$

  – welche Lieferanten liefern alle Teile?

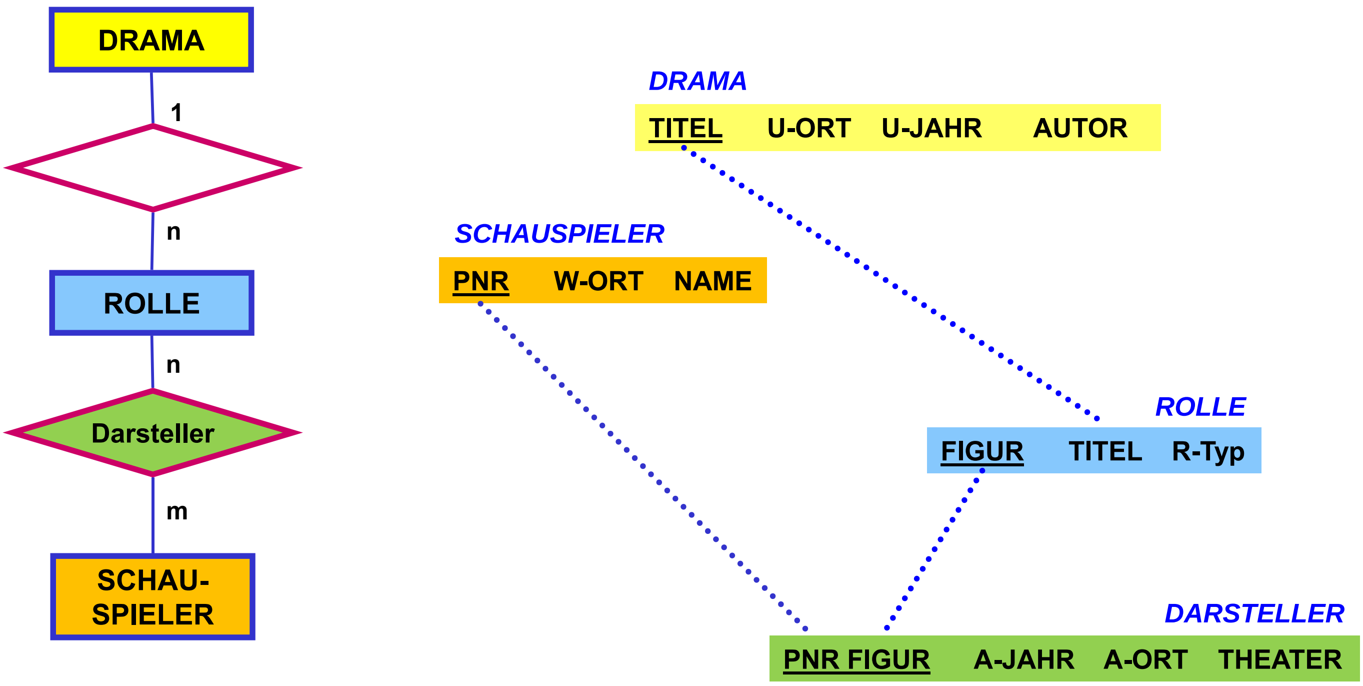
$$\pi_{LNR,TNR}\ (LPT) \div \pi_{TNR}\ (PT)$$

- Zusammenhang zwischen Division und kartesischem Produkt:

$$(\ R \times S\ )\ \div\ S\ =\ R$$

$$(\ R \div S\ )\ \times\ S\ =\ R\ ?$$

Beispiel-DB: Bühne



Beispielanfragen

- welche Darsteller (PNR) haben im Schauspielhaus gespielt?

  π_{PNR} ($\sigma_{THEATER='Schauspielhaus'}$ (DARSTELLER))

- welche Schauspieler (NAME, W-ORT) haben im 'Faust' mitgespielt?

  $\pi_{NAME,W\text{-}ORT}$ ($\sigma_{TITEL='Faust'}$ (SCHAUSPIELER $\bowtie$ (DARSTELLER $\bowtie$ ROLLE)))

- Finde alle Schauspieler (NAME), die bei in Weimar uraufgeführten Dramen an ihrem Wohnort als 'Held' mitgespielt haben

π_{NAME} ($\sigma_{U\text{-}ORT='Weimar' \wedge R\text{-}Typ='Held' \wedge W\text{-}ORT=A\text{-}ORT}$ (SCHAUSPIELER $\bowtie$ (DARSTELLER $\bowtie$ (ROLLE $\bowtie$ DRAMA))))

- Finde die Schauspieler (PNR), die nie gespielt haben

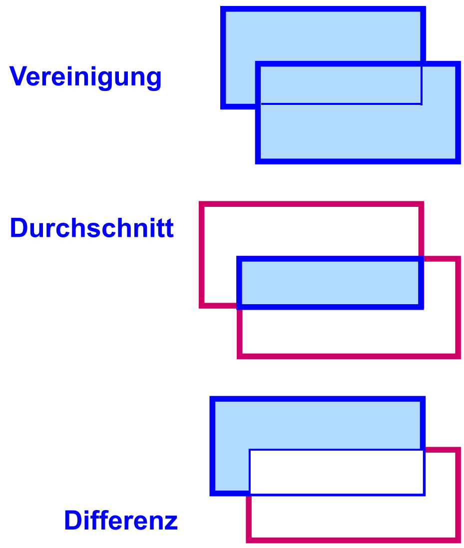
  π_{PNR} (SCHAUSPIELER) -

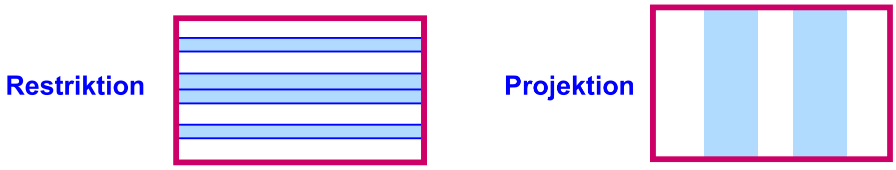
- Finde die Schauspieler (NAME), die alle Rollen gespielt haben

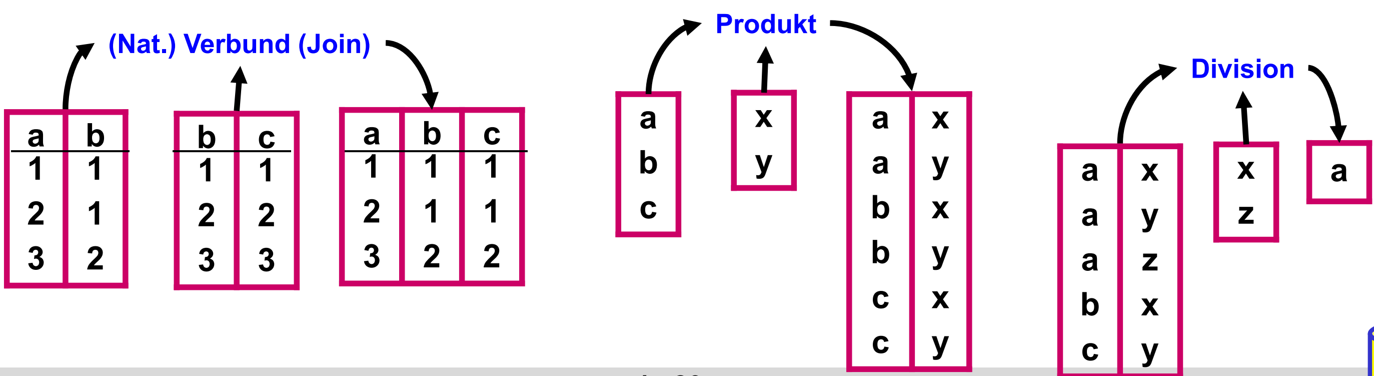
  π_{NAME} (SCHAUSPIELER $\bowtie$

Zusammenfassung Relationenalgebra

- saubere mathematische Definition
- mengenorientierte Operationen
- keine Änderungsoperationen!
- für Laien nicht leicht verständlich







a	b
1	1
2	1
3	2

b	c
1	1
2	2
3	3

a	b	c
1	1	1
2	1	1
3	2	2

a
b
c

x
y

a	x
a	y
b	x
b	y
c	x
c	y

a	x
a	y
a	z
b	x
c	y

x
z

a