4. RELATIONENALGEBRA

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

SPRACHEN FÜR DAS RELATIONENMODELL

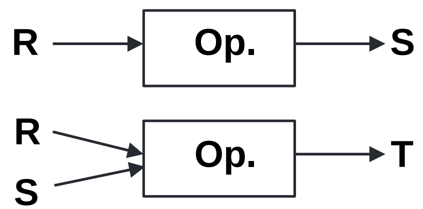
- Datenmodell = Datenobjekte + Operatoren
- im RM wird vereinheitlichte Sprache angestrebt für:
  - Anfragen (Queries) im 'Stand-Alone'-Modus
  - Datenmanipulation und Anfragen eingebettet in eine Wirtssprache
  - Datendefinition
  - Zugriffs- und Integritätskontrolle
  - Unterstützung verschiedener 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	Relationenoperationen
• Vereiningung • Differenz • Kartesisches Produkt • Durchschnitt	• Restriktion (Selektion) • Projektion • Umbenennung • 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 $q \in \{<, =, >, \leq, \neq, \geq\}$
  - logische Operatoren: $\vee$, $\wedge$, $\neg$
- Beispiele:
  - $\sigma_{SALARY < BONUS}$ (PERS)
  - $\sigma_{OCCUPATION='Programmer' \wedge AGE < 50}$ (PERS)
- Eigenschaften
  - $grad(\sigma_P(R)) = grad(R)$
  - $card(\sigma_P(R)) \leq card(R)$

PROJEKTION

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

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

- Beispiel: $\pi_{\text{Name, Salary}}(\text{PERS})$
- Eigenschaften:
  - wichtig: Duplikate werden entfernt ! (Mengeneigenschaft)

    z.B. für $\pi_{\text{DNO}}(\text{PERS})$

  - grad (π_A(R)) <= grad (R)
  - card (π_A(R)) <= card (R)

UMBENNUNG

- Umbennung von Attributnamen und Relationsnamen einer Relation $R(B_1,\ldots B_n)$
  - Wichtig zum Auflösen von Mehrdeutigkeiten bei reflexiven Verbünden für eindeutige Adressierung des Attributs

$\rho_{S(A_1,\ldots,A_n)}(R) \Rightarrow$ Umbenennung der Attribute und des Namens von R zu $S(A_1, \ldots A_n)$

$\rho_S(R) \Rightarrow$ Umbenennung der Relation R zu S

- Beispiel: $\rho_{MA(Nachname,Gehalt)}(\pi_{Name,\ Salary}(PERS))$

RELATIONENALGEBRA: BEISPIEL-DB

DEPT

DNO	DNAME	CITY
K51	Planung	Leipzig
K53	Einkauf	Frankfurt
K55	Vertrieb	Frankfurt

PERS

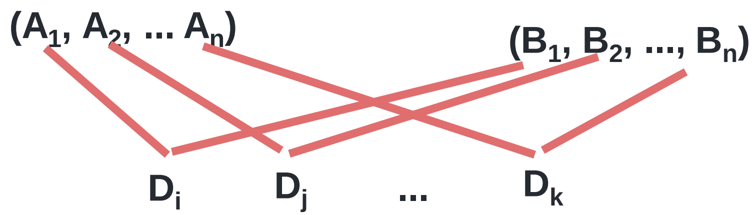
PNO	NAME	AGE	SALARY	DNO(FS auf DEPT)	MGR(FS auf PERS)
406	Abel	47	50700	K55	123
123	Schulz	32	43500	K51	-
829	Müller	36	40200	K53	406
574	Schmid	28	36000	K55	123

- Finde alle Angestellten aus Abteilung K55, die mehr als 40.000 verdienen
- $\sigma_{\text{DNO='K55'} \wedge \text{SALARY} > 40000}\ (\text{PERS})$
- Finde alle Abteilungsorte
- $\pi_{\text{CITY}}\ (\text{DEPT})$
- Finde den Abteilungsnamen von Abteilung K53
- $\pi_{\text{DNAME}}\ (\sigma_{\text{DNO='K53'}}(\text{DEPT}))$

CITY
Leipzig
Frankfurt

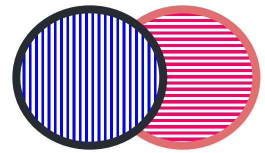
KLASSISCHE MENGENOPERATIONEN

– Voraussetzung: ***Vereinigungsverträglichkeit*** der beteiligten Relationen:



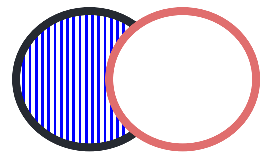
  – gleicher Grad und gleiche Bereiche:
  ➢ $W(A_i) = W(B_i) \quad : \quad i = 1, n$

– Vereinigung: $R \cup S = \{ t \mid t \in R \vee t \in S \}$



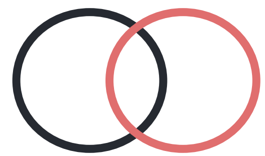
  – $\text{card}(R \cup S) <= \text{card}(R) + \text{card}(S)$

– Differenz: $R - S = \{ t \mid t \in R \wedge t \notin S \}$



  – $\text{card}(R - S) <= \text{card}(R)$

– Durchschnitt:
$R \cap S = R - (R - S) = \{ t \mid t \in R \wedge t \in S \}$



  – $\text{card}(R \cap S) \leq \min(\text{card}(R), \text{card}(S))$

– Beispielanfrage: Welche Abteilungen (DNO) haben keine Mitarbeiter?

(ERWEITERTES) KARTESISCHES PRODUKT

– $R(A_1, \ldots A_r)$ (Grad r) und $S(B_1, \ldots B_s)$ (Grad s) beliebig

$$\boxed{\begin{array}{l} \text{Relationsschema: } (R \times S)(A_1, \ldots, A_r, B_1, \ldots, B_s) \\ \text{Relation: } R \times S = \{k = x \circ y \mid x \in R \wedge y \in S\} \end{array}}$$

– Beachte: $k = x \circ y = (x_1, \ldots, x_r, y_1, \ldots, y_s)$
  nicht $((x_1, \ldots, x_r), (y_1, \ldots, y_s))$ wie übliches kart. Produkt
– $\text{grad}(R \times S) = \text{grad}(R) + \text{grad}(S)$
– $\text{card}(R \times S) = \text{card}(R) \cdot \text{card}(S)$

Beispiel

R

A	B	C
a	g	1
d	a	2
b	b	3

S

D	E	F
b	g	3
d	a	2

R × S

A	B	C	D	E	F
a	g	1	b	g	3
a	g	1	d	a	2
d	a	2	b	g	3
d	a	2	d	a	2
b	b	3	b	g	3
b	b	3	d	a	2

VERBÜNDE

- Operatoren für die Verknüpfung von Tupeln $r \in R$ und s $\in S$ unter Berücksichtigung eines Verbundprädikats P
- Allgemeiner Verbund – Theta Join
  - Spezialfall Gleichverbund → Equi-Join
- Natürlicher Verbund – Natural Join
  - Equi Join über gleichnamige Attribute
- [linker, rechter] Semi-Join
  - Erhaltung der Attribute einer Relation
- Verlustfreie Verbünde
  - Erhalt der Tupel auch ohne Verbundpartner
  - [linker, rechter] Äußerer Verbund

ALLGEMEINER VERBUND (THETA-JOIN)

- grob: kartesisches Produkt zwischen zwei Relationen R und S.
  - eingeschränkt durch Θ -Bedingungen o.B.d.A. zwischen Attribut A_k von R und Attribut B_l von S → verallgemeinerbar für mehrere Attribute

Relationsschema: $(R \bowtie_{A_k \Theta B_l} S)\ (A_1, \ldots, A_r, B_1, \ldots, B_s)$

Relation: $(R \bowtie_{A_k \Theta B_l} S) = \{k = x \circ y \mid x \in R \land y \in S \land R: x[A_k]\ \Theta\ y[B_l]\}$

- Θ-Verbund zwischen R und S ableitbar:

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

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

  - grad $(R \bowtie_{A\Theta B} S)$ = grad (R x S) = grad(R) + grad (S)
  - card $(R \bowtie_{A\Theta B} S) \leq$ card (R x S)
- für häufigen Fall des ***Gleichverbunds (Equi-Join)*** gilt Θ = ’=’ :

NATÜRLICHER VERBUND (NATURAL JOIN)

– grob: Gleichverbund über alle 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): $B_1 = A_{r-j+1}$, $B_2 = A_{r-j+2}$ … $B_j = A_r$

$$R \bowtie S = \pi\big(A_1, A_2, \ldots, A_r, B_{j+1}, \ldots, B_s\big)\sigma_{A_{r-j+1}=B_1 \wedge \cdots \wedge A_r=B_j}(R \times S)$$

$\bowtie$ Zeichen für Natural Join $\Rightarrow$ **Θ = '='**

– Join-Attribute sind durch Übereinstimmungsbedingung gegeben

– $\mathrm{grad}(R \bowtie S) = \mathrm{grad}(R) + \mathrm{grad}(S) - j$

  – mit j Anzahl der gemeinsamen Attribute

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

DEPT

<u>DNO</u>	DNAME	CITY
K51	Planung	Leipzig
K53	Einkauf	Frankfurt
K55	Vertrieb	Frankfurt

PERS

<u>PNO</u>	NAME	AGE	SALARY	DNO(FS auf DEPT)	MGR(FS auf PERS)
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 (PNO, AGE, DNAME), die in einer Abteilung in Frankfurt arbeiten und älter als 30 sind
- $\pi_{PNO,AGE,DNAME}(\sigma_{City='Frankfurt' \wedge AGE>30}(DEPT \bowtie PERS))$

PNO	AGE	DNAME
406	47	Vertrieb
829	36	Einkauf

SEMI-JOIN

- Ergebnisbeschränkung des Gleichverbundes auf eine der beiden Eingaberelationen R und S

$$\mathrm{R} \ltimes S = \pi_{R-Attribute}(R \bowtie S) \qquad \text{linker Semi-Join}$$

$$\mathrm{R} \rtimes S = \pi_{S-Attribute}(R \bowtie S) \qquad \text{rechter Semi-Join}$$

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

$\rtimes$

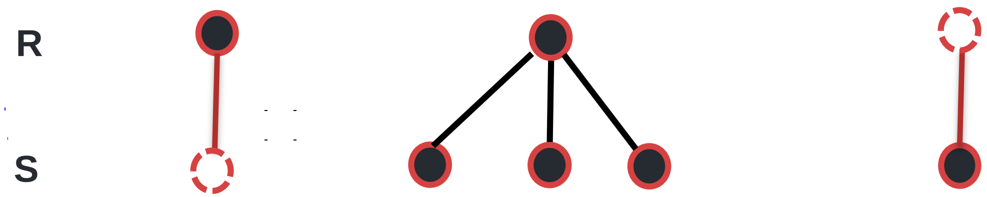
S		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

=

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).
- $R \bowtie S$ verlustfrei ⇔ $\pi_{\mathrm{R-Attribute}} R \ltimes S = R \wedge \pi_{\mathrm{S-Attribute}} R \rtimes S = S$
- bisher: R 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, {≡} der undefinierte Wert

$$R' := R \cup ((\pi_A(S) - \pi_A(R)) \times \{ \underbrace{(\equiv, \ldots, \equiv)}_{grad(R) - grad(\pi_A(R))} \}$$

$$S' := S \cup ((\pi_A(R) - \pi_A(S)) \times \{ \underbrace{(\equiv, \ldots, \equiv)}_{grad(S) - grad(\pi_A(S))} \}$$

äußerer Gleichverbund
$R ⟗_{R.A=S.A} S := R' \bowtie_{R'.A=S'.A} S'$

äußerer natürlicher Gleichverbund
$\mathrm{R} ⟗ \mathrm{S} := \mathrm{R}' \bowtie S'$

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

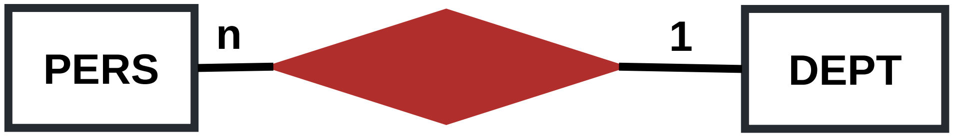
linker äußerer Gleichverbund
$R ⟕ S := R \bowtie S'$

rechter äußerer Gleichverbund
$\mathrm{R} ⟖ \mathrm{S} := \mathrm{R}' \bowtie_{\mathrm{R}'.\mathrm{A}=\mathrm{S.B}} \mathrm{S}$

verallgemeinerbar auf 2 (oder mehr) Joins, z.B. $\mathrm{R} ⟗ \mathrm{S} ⟗ \mathrm{T}$

- selbst isolierte Tupel können zu einem vollständigen Pfad expandiert werden

OUTER JOIN - BEISPIEL



PERS

PNO	DNO ...
P1	D1
P2	D1
P3	D2
P4	-
P5	-

DEPT

DNO	DNAME ...
D1	A
D2	B
D3	C

PERS ⋈ DEPT

PNO	DNO	DNAME ...
P1	D1	A
P2	D1	A
P3	D2	B

PNO	DNO	DNAME ...
P1	D1	A
P2	D1	A
P3	D2	B

PERS⟗DEPT

PERS ⟖ DEPT

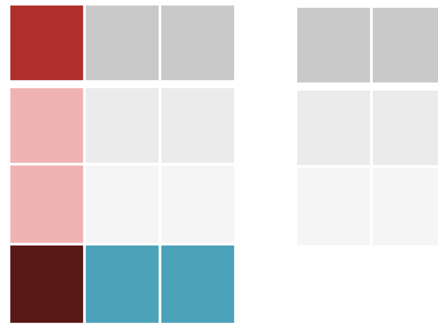
PNO	DNO	DNAME ...
P1	D1	A
P2	D1	A
P3	D2	B

PNO	DNO	DNAME ...
P1	D1	A
P2	D1	A
P3	D2	B

PERS⟕DEPT

DIVISION

- Beantwortung von Fragen, bei denen eine „ganze Relation“ zur Qualifikation herangezogen wird
- Simulation des Allquantors $\Rightarrow$ eine Attributwert-Kombination aus R steht mit allen Tupeln aus S in einer bestimmten Beziehung
- <u>**Definition**</u>



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

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

$\text{card}(R \div S) \le \text{card}(R)$

DIVISION (2)

– Beispiel

SUPPLY

SNR	PRO	PART
L1	P1	T1
L1	P2	T1
L2	P1	T1
L2	P1	T2
L2	P2	T1

÷

PP

PRO	PART
P1	T1
P1	T2
P2	T1

=

SNR
L2

– welche Lieferanten beliefern alle Projekte?

$$\pi_{SNR,PRO}\ (SUPPLY) \div \pi_{PRO}\ (PP)$$

– welche Lieferanten liefern alle Teile?

$$\pi_{SNR,PART}\ (SUPPLY) \div \pi_{PART}\ (PP)$$

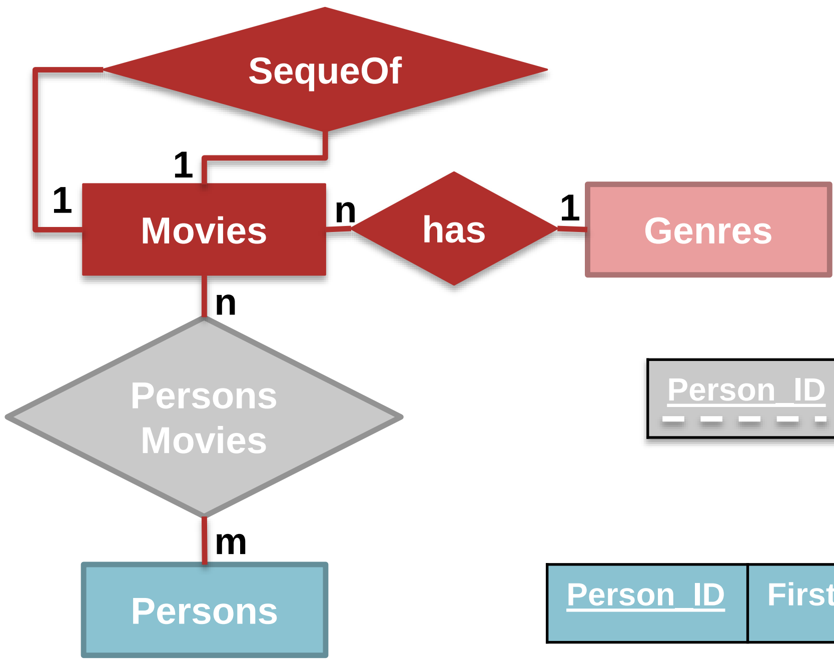
– Zusammenhang zwischen Division und kartesischem Produkt:

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

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

BEISPIEL-DB: DBS1 MOVIEDB

Movie_ID	Title	Release Date	Genre_ID	Budget	Opening Week	Profit	Runtime	Certificate	Sequel Of	Distribution



Genre_ID	Name

Person_ID	Movie_ID	Role

Person_ID	Firstname	Lastname

https://dbis-uibk.github.io/relax/calc/gist/d37f667154aec34f5c4954723ae01db9/DBS1_MovieDB/0

BEISPIELANFRAGEN

- Welche Filme haben Überlänge (Runtime> 120)?

$\pi_{\text{Title}} (\sigma_{\text{Runtime} > 120} (\text{Movies}))$

- Welche Personen (Firstname, Lastname) waren an Filmen des Genres ‚Fantasy' beteiligt?

$\pi_{\text{Firstname, Lastname}}\ \sigma_{\text{Genres.Name='Fantasy'}}(\text{Genres} \bowtie \text{Movies} \bowtie \text{PersonsMovies} \bowtie \text{Persons})$

- Finde alle Filme (Title), wo mindestens 2 Personen die Regie (Role = “direction“) geführt haben.
- $\rho_{P2} (\text{PersonsMovies}) \bowtie_{\text{P2.Movie\_ID = P1.Movie\_ID} \wedge \text{P1.Person\_ID} \neq \text{P2.Person\_ID}}$
  $\rho_{P1} (\text{PersonsMovies})$
- Finde die Genres (Name), zu denen kein Film existiert

$\pi_{\text{Genre\_ID}} (\text{Genres})\ -$

- Welche Personen (Person_ID) waren an allen Filmen des Genres ‚Drama' beteiligt?

ZUSAMMENFASSUNG RELATIONENALGEBRA

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