
Įvadas į reliacinę algebrą ir SQL

PIRMA DALIS

Reliacinis DB modelis

Reliacinį modelį 1970 m. pasiūlė mokslininkas *E. F. Codd*.

Reliacinis duomenų bazės modelis (angl. *Relational DB model*) – matematinis duomenų modelis, skirtas reliacinėms DB projektuoti.

Reliacinis DB modelis apima:

- Loginę duomenų struktūrą, išreikštą susijusiais santykiais
- Duomenų darnumą užtikrinančią integralumo taisyklių aibę
- **Operacijų aibę, skirta manipuluoti duomenimis – reliacinę algebrą**

Reliacinė algebra (RA) ir SQL

SQL (ang. *Structured Query Language*) – kalba užklausoms ir modifikacijoms RDB-je atlikti.

SQL - programinė reliacinės algebros realizacija.

Trumpa *SQL* priešistorė:

- Pirmoji versija: 1970 m. kompanija *IBM, System R*
- Autoriai: *Donald D. Chamberlin* ir *Raymond F. Boyce*
- *Pirmasis pavadinimas: SEQUEL (Structured English QUery Language)*

<http://patorjk.com/blog/2012/01/26/pronouncing-sql-s-q-l-or-sequel/>

SQL savybės

Deklaratyvi, procedūrinė kalba

Apibrėžta standarto ISO/IEC 9075

Yra daug dialektų

Kalbą sudaro 3 dalys:

- Duomenų deklaravimo kalba (*Data Definition Language (DDL)*)
- **Duomenų manipuliavimo kalba** (*Data Manipulation Language (DML)*)
- Duomenų kontrolės kalba (*Data Control Language (DCL)*)

SQL standarto versijų istorija

Vardas	Paaiškinimai
SQL-86	Pirmasis ANSI standartas.
SQL-89	Papildyta integralumo apribojimais.
SQL-92	ISO 9075 standartas.
SQL:1999	Papildytas reguliariomis išraiškomis, rekursinėmis užklausomis, trigeriais ir kt.
SQL:2003	Papildymai susiję su XML naudojimu (SQL/XML), standartizuotos sekos ir automatiškai generuojamos stulpelių reikšmės.
SQL:2006	Apibrėžė, kaip DB galima saugoti XML duomenis, manipuluoti jais, publikuoti XML formoje ir kt.
SQL:2008	Papildytas naujomis savybėmis, pvz., INSTEAD OF trigeriai, TRUNCATE sakiny.
SQL:2011	Pritaikyta laiko (angl. <i>temporal</i>) DB

SQL realizacija DBVS

Skirtingų gamintojų SQL realizacijos tarpusavyje nesuderintos

Gamintojas nebūtinai laikosi SQL standarto

- *PostgreSQL* - labiausiai besilaikanti standarto DBVS

Dažniausi skirtumai:

- Datos ir laiko formatai
- Simbolių eilučių jungimas
- *NULL* reikšmių naudojimas
- Jautrumas raidžių dydžiui lyginant reikšmes

Dažniausios SQL standarto nesilaikymo priežastys:

- Standarto dydis ir sudėtingumas
- Standartas neapibrėžia kaip turi elgtis DBVS tam tikrose srityse, pvz., indeksuojant, saugant failus
- Tiksliai apibrėžia kalbos sintaksę, bet ne semantiką (prasmę)
- Dėl klientų naudojamų DB gamintojai nelinekę daryti radikalių pakeitimų savo DBVS pasikeitus standartui
- Gamintojų noras pririšti prie produkto klientą
- Išskirtinių, su standartu nesuderinamų, savybių turėjimas, pvz., greیتaveika.

RA operacijos ir jų realizacija *SQL*

Kodėl multi-aibės?

- Reliacinės algebros operacijos su multi-aibėmis atliekamos greičiau, nes **nereikia tikrinti**, ar gautas santykio kortežas nedubliuoja prieš tai gautų kortežų.
- Jei atlikę projekciją norima gauti agregatą, pavyzdžiui, tam tikrų stulpelių reikšmių vidurkį arba sumą, tai eliminuoti besidubliuojančius kortežus **negalima**.

Reliacinės algebros operacijos

Pagrindinės operacijos:

- Projekcija
- Selekcija
- Suma
- Skirtumas
- Sandauga

Papildomos (išvestinės) operacijos:

- Sankirta
- Jungtis
- Natūralioji jungtis

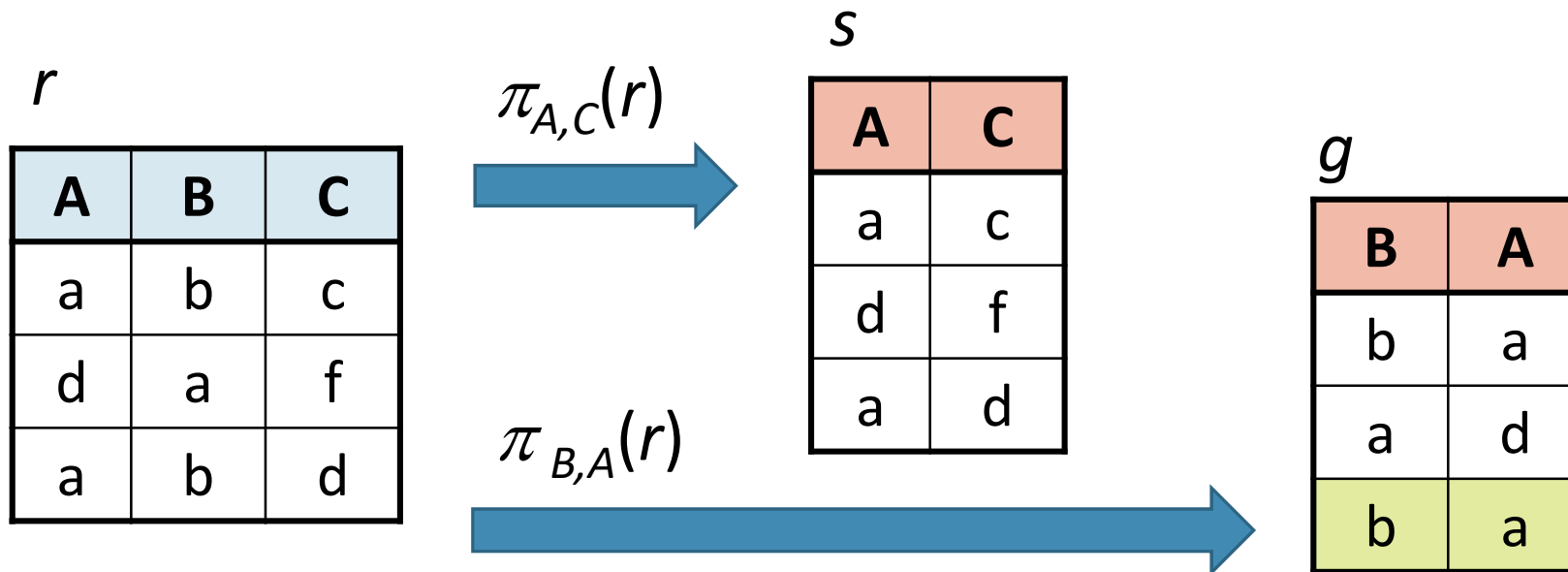
Projekcija (angl. *Projection*)

Apibrėžta schema $\mathbf{R}=(A, B, C)$

Duotas schemas \mathbf{R} santykis r

Santykio r projekcija schemoje $\mathbf{S}=(A, C)$ bus santykis $s=\pi_{A,C}(r)$

Santykio r projekcija schemoje $\mathbf{S}=(B, A)$ bus santykis $g=\pi_{B,A}(r)$

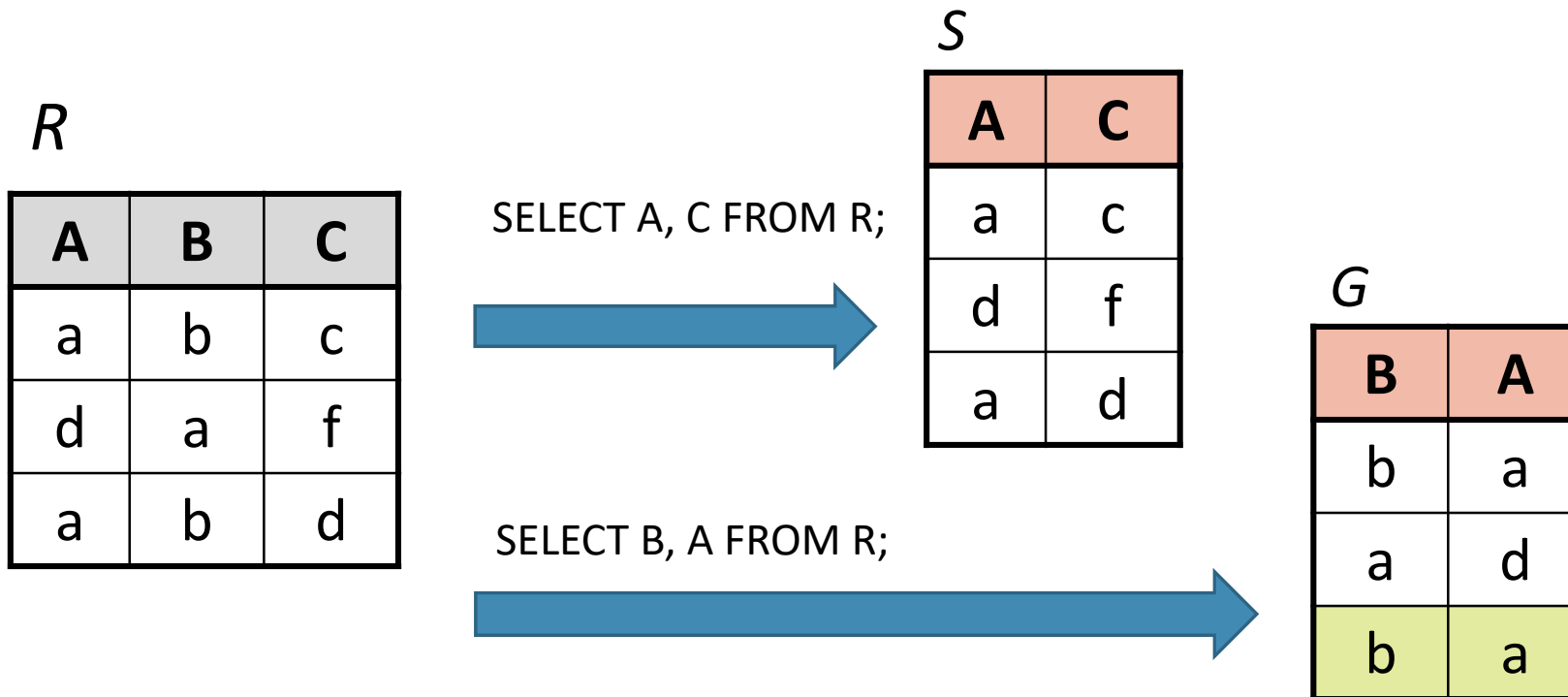


Projekcijos realizacija SQL

Sukurta lentelė **R**, kurios schema **R=(A, B, C)**

$\pi_{A,C}(r)$ realizuojanti SQL užklausa bus **SELECT A, C FROM R;**

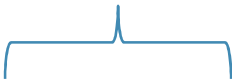
$\pi_{B,A}(r)$ realizuojanti SQL užklausa bus **SELECT B, A FROM R;**



Projekcija

SELECT ...

π



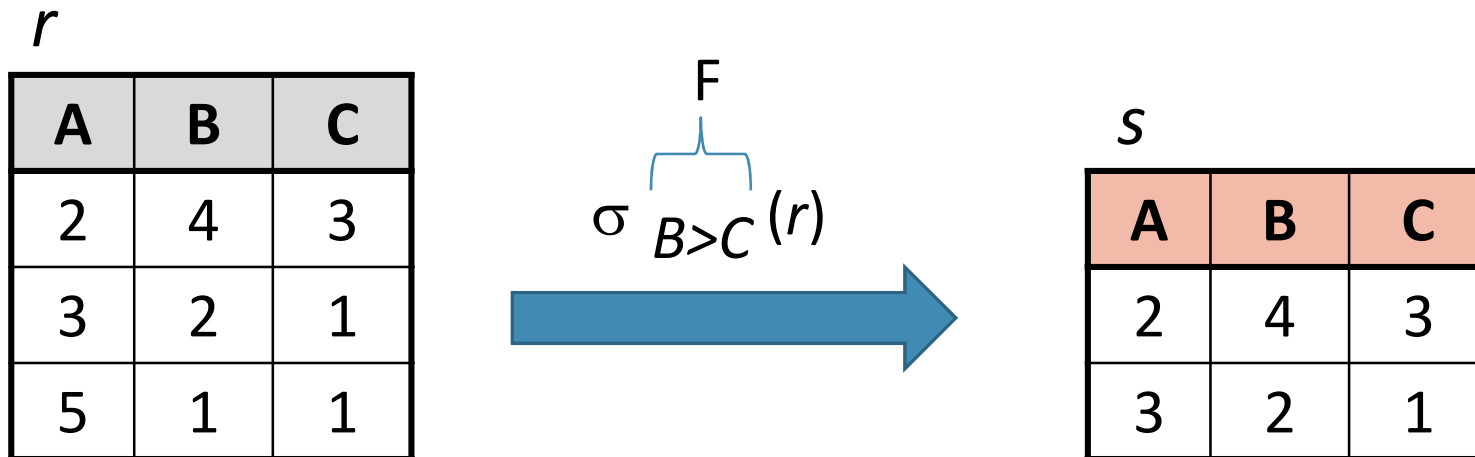


Selekcija (angl. *Selection*)

Sakykime F – tai formulė sudaryta iš:

- Operandų, kuriais gali būti konstantos ir/arba atributai $A_i \in R$
- Aritmetinių palyginimo operatorių ($=, \neq, <, \leq, >, \geq$)
- Loginių operatorių (\wedge, \vee, \neg)

Selekcijos $\sigma_F(r)$ rezultatas – santykis s , kurį sudaro aibė tokių r priklausančių kortežų t , kad įrašius konstantas ir /arba kortežo t komponentus $t(A_i)$ į formulės F atitinkamus operandus, formulė gaunama teisinga.



Selekcijos realizacija SQL

Sukurta lentelė **R**, kurios schema **R=(A, B, C)**

$\sigma_{B>C}(r)$ realizuojanti SQL užklausa bus

SELECT A, B, C FROM R WHERE B>C;

R

A	B	C
2	4	3
3	2	1
5	1	1

SELECT A, B, C FROM R
WHERE B>C;



S

A	B	C
2	4	3
3	2	1

Selekcija

**SELECT ...
FROM ...
WHERE ...**

σ



Paprasta SQL užklausa

Būtina
dalis

```
SELECT <stulpeliai>  
FROM <lentelės>  
[WHERE <sąlyga>];
```

Užklauso rezultato struktūra – per kablelį išvardinti lentelių stulpeliai (ir išvestiniai), kurių reikšmės turi būti pateiktos atsakyme ir tokia tvarka kaip išvardinta

Per kablelį išvardintos arba su jungimo operatoriumi sujungtos lentelės, kurios dalyvauja užklausoje

Sąlyga, kurią turi tenkinti visos atsakymo eilutės.

Sąlygą gali sudaryti:

- Stulpeliai
- Konstantos
- Loginiai operatoriai
- Kintamieji
- Funkcijos
- Algebriniai operatoriai
- Specialūs SQL operatoriai

SQL užklausoos pavyzdys

*Išrink vardą ir pavardę klientų,
kurie gimė vėliau nei 1975
metais.*

SELECT vardas, pavarde

FROM klientai

WHERE year(gimimo_data) > 1975;

KLIENTAI

ASMENS_KODAS	VARDAS	PAVARDE	GIMIMO_DATA	TELEFONAS
48010038888	Rūta	Rainytė	1980-10-03	866661116
37811139999	Linas	Lapkus	1978-11-13	867771117
37112254444	Tomas	Tomkus	1971-12-25	869991119
46707172222	Agnė	Austienė	1967-07-17	888881118

SQL užklausos pavyzdys. Atsakymas

*Išrink vardą ir pavardę
klientų, kurie gimė
vėliau nei 1975 metais.*

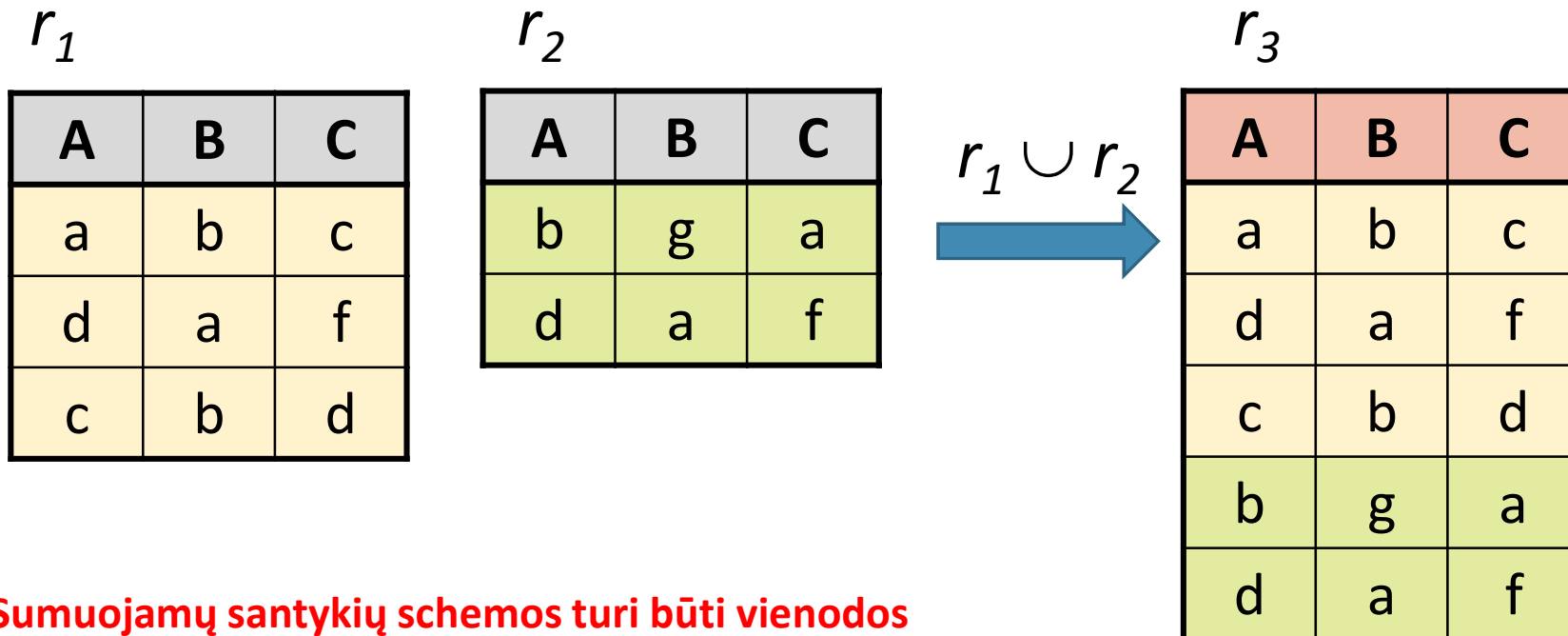
```
SELECT vardas, pavarde  
FROM klientai  
WHERE year(gimimo_data)>1975;
```

Atsakymas:

VARDAS	PAVARDE
Rūta	Rainytė
Linas	Lapkus

RA. Suma (angl. *Union*)

Schemas **R** santykių r_1 ir r_2 suma $r_3 = r_1 \cup r_2$ yra aibė (multi) kortežų, kurie priklauso arba santykiui r_1 , arba r_2 , arba abiem kartu:



Sumuojamų santykių schemas turi būti vienodos

Sumos realizacija SQL

KTU

PAVARDE	VARDAS	APGYNE
Petraitis	Petras	1990
Jonaitis	Jonas	1995
Antanaitis	Antanas	1990

VU

PAV	VAR	APG
Petraitis	Petras	1990
Zuokys	Jonas	1995
Antanaitis	Antanas	1993

KTU VU

PAVARDE	VARDAS	APGYNE
Petraitis	Petras	1990
Jonaitis	Jonas	1995
Antanaitis	Antanas	1990
Petraitis	Petras	1990
Zuokys	Jonas	1995
Antanaitis	Antanas	1993

$\pi_{\text{PAVARDE, VARDAS, APGYNE}} ((\text{KTU}) \cup (\text{VU}))$

(SELECT pavarde, vardas, apgyne
FROM ktu)

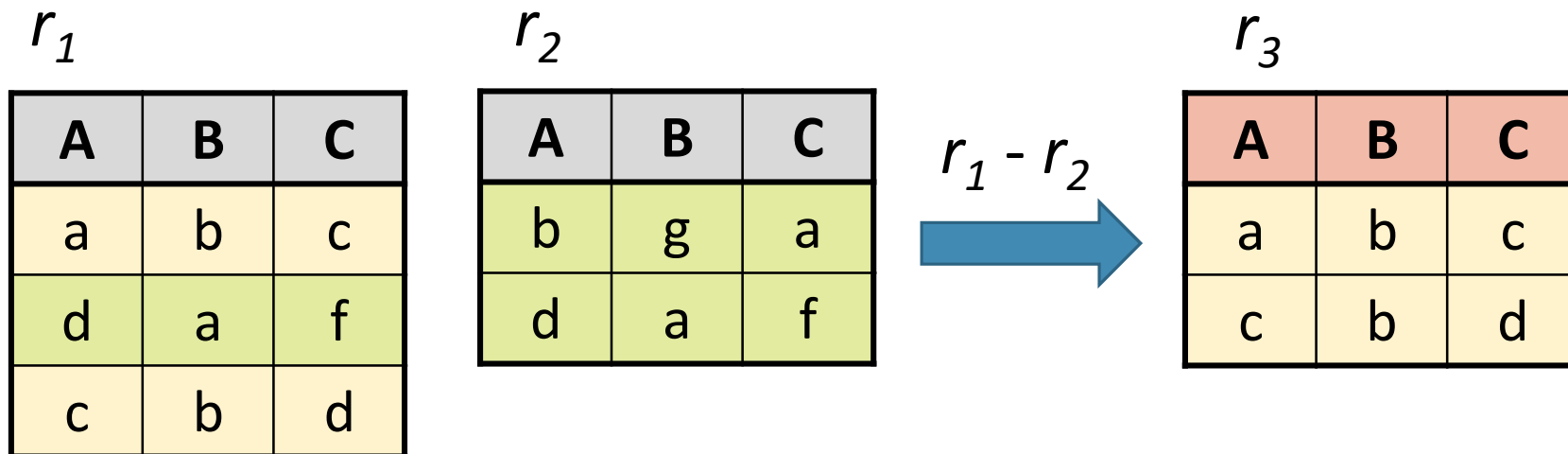
UNION

(SELECT pav AS pavarde, var AS
vardas, apg AS apgyne FROM vu) ;

Sumuojamų lentelių schemas turi būti vienodos

RA. Skirtumas (angl. *Difference*)

Schemas **R** santykių r_1 ir r_2 skirtumas $r_3 = r_1 - r_2$ yra aibė kortežų, kurie priklauso santykiui r_1 , bet nepriklauso r_2 :



Santykių schemas turi būti vienodos

Skirtumo realizacija SQL

KTU

PAVARDE	VARDAS	APGYNE
Petraitis	Petras	1990
Jonaitis	Jonas	1995
Antanaitis	Antanas	1990

VU

PAVARDE	VARDAS	APGYNE
Petraitis	Petras	1990
Zuokys	Jonas	1995
Antanaitis	Antanas	1993

Užklausa: Kurie asmenys studijavo tik KTU . Pateikite tik asmenų pavardes ir vardus.

TIK KTU

PAVARDE	VARDAS
Jonaitis	Jonas

$\pi_{\text{PAVARDE, VARDAS}} ((\text{KTU}) - (\text{VU}))$

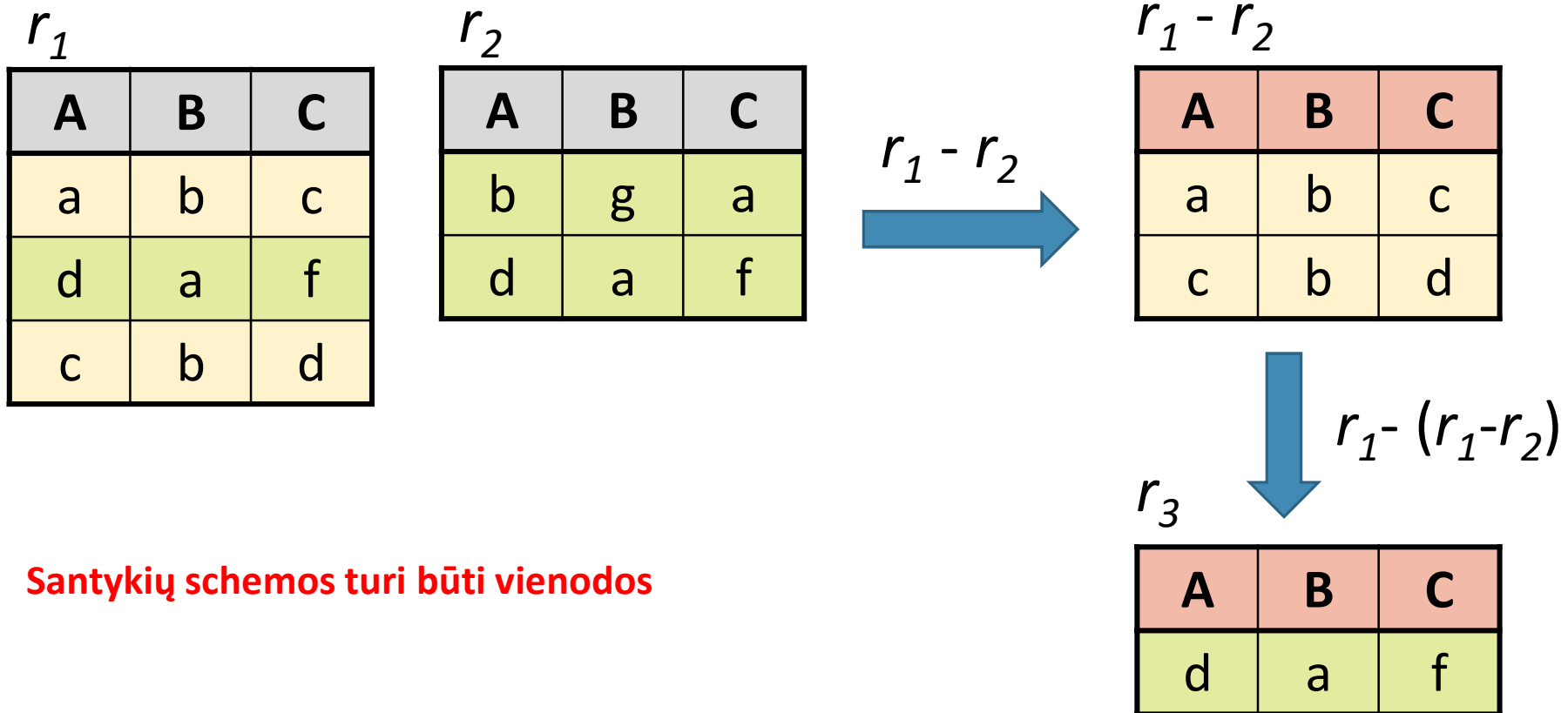
(SELECT pavarde, vardas FROM ktu)
EXCEPT
 (SELECT pavarde, vardas FROM vu) ;

Lentelių schemas turi būti vienodos

Ne visos DBVS (tam tarpe MySQL) realizuoja šią operaciją.

RA. Sankirta (angl. *Intersection*)

Schemas **R** santykių r_1 ir r_2 sankirta $r_3 = r_1 \cap r_2$ yra aibė kortežų, kurie priklauso ir santykiui r_1 , ir santykiui r_2 . Sankirta r_3 nustatoma du kartus panaudojus skirtumo operaciją $r_1 - (r_1 - r_2)$:



Santykių schemas turi būti vienodos

Sankirtos realizacija SQL

KTU

PAVARDE	VARDAS	APGYNE
Petraitis	Petras	1990
Jonaitis	Jonas	1995
Antanaitis	Antanas	1990

VU

PAVARDE	VARDAS	APGYNE
Petraitis	Petras	1990
Zuokys	Jonas	1995
Antanaitis	Antanas	1993

Užklausa: Kurie asmenys studijavo ir KTU, ir VU . Pateikite tik asmenų pavardes ir vardus.

$$\pi_{\text{PAVARDE, VARDAS}} ((\text{KTU}) \cap (\text{VU}))$$

KTU IR VU

PAVARDE	VARDAS
Petraitis	Petras
Antanaitis	Antanas

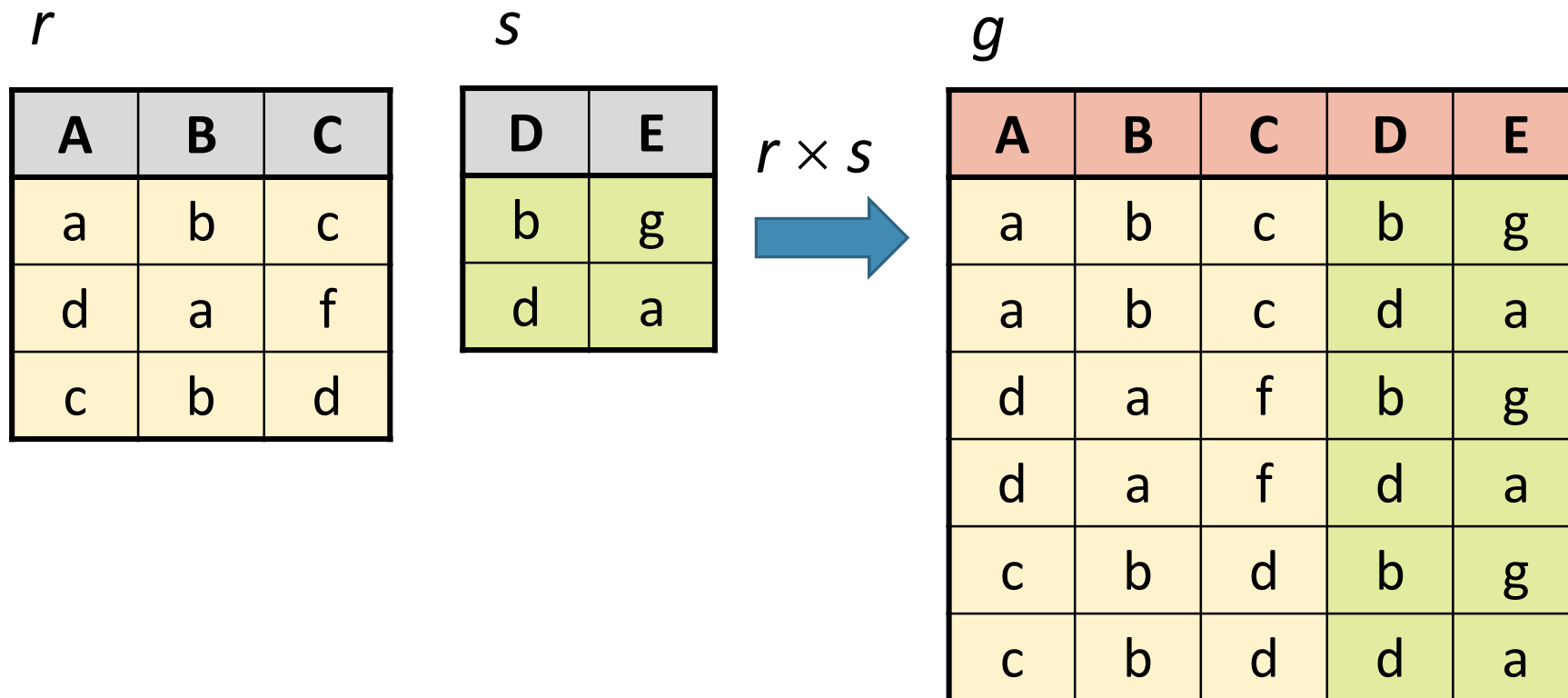
(SELECT pavarde, vardas FROM ktu)
INTERSECT
 (SELECT pavarde, vardas FROM vu) ;

Lentelių schemas turi būti vienodos

Ne visos DBVS (tam tarpe MySQL) realizuoja šią operaciją.

RA. Sandauga (angl. *Cartesian product*)

Dviejų schemų **R** ir **S** santykių r ir s sandauga $g=r \times s$ yra schemos $G=R \otimes S$ kortežų aibė, kuri gaunama iš santykio r kortežų visų kombinacijų su visais santykio s kortežais:



Sandaugos realizacija SQL

STUD

PAVARDE	VARDAS	APGYNE
Petraitis	Petras	1990
Jonaitis	Jonas	1995
Antanaitis	Antanas	1990

DAL

KODAS	PAVADINIMAS
T000B200	Matematika
T000B300	Fizika

SELECT pavarde, vardas,
apgyne, kodas, pavadinimas
FROM stud, dal ;

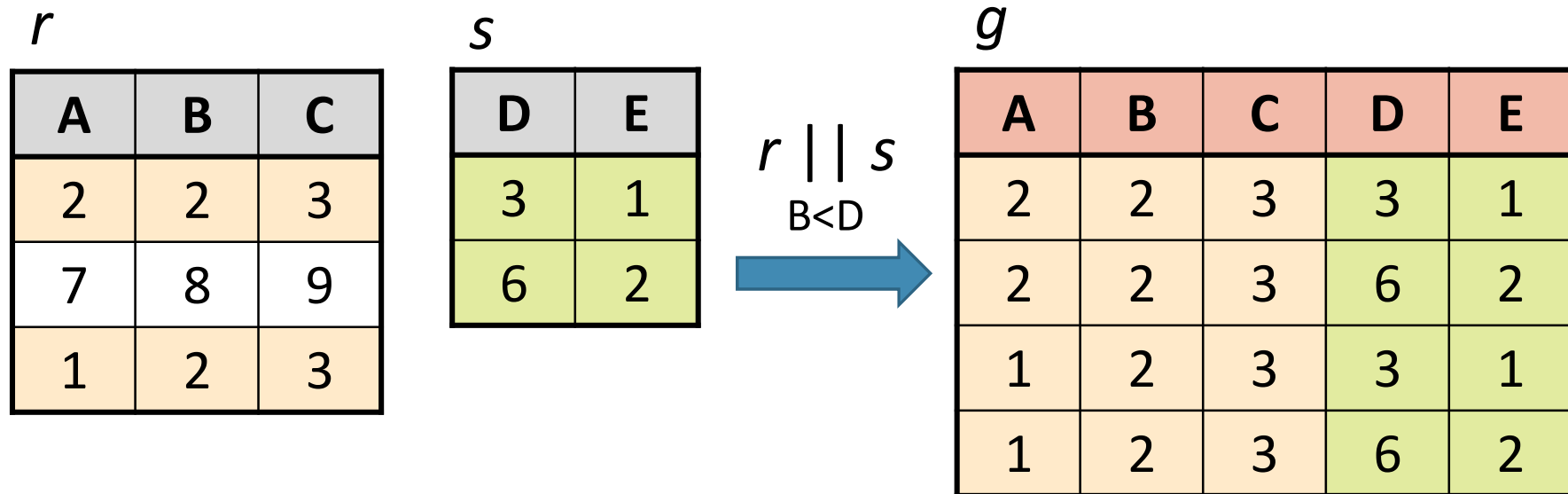
STUD × DAL

STUD DAL

PAVARDE	VARDAS	APGYNE	KODAS	PAVADINIMAS
Petraitis	Petras	1990	T000B200	Matematika
Jonaitis	Jonas	1995	T000B200	Matematika
Antanaitis	Antanas	1990	T000B200	Matematika
Petraitis	Petras	1990	T000B300	Fizika
Jonaitis	Jonas	1995	T000B300	Fizika
Antanaitis	Antanas	1990	T000B300	Fizika

RA. (Teta) Jungtis (angl. *Theta-Joins*)

Schemų **R** ir **S** santykių r ir s jungtis stulpelių $A_i \in R$ ir $B_j \in S$ atžvilgiu panaudojus aritmetinį palyginimo operatorių Θ , užrašoma $r \bowtie s$. Operatoriumi Θ gali būti $=, \neq, <, \leq, >, \geq$. Atlikus operaciją, gaunamas schemas $G=(R \cup S)$ santykis g , kuris yra toks dekartinės sandaugos $r \times s$ poaibis, kad kiekviename jo korteže t komponentas $t(A_i)$ turi ryšį su to paties kortežo komponentu $t(B_j)$:



(Teta) Jungties realizacija SQL

GAMINIAI

KODAS	KIEKIS	SVORIS
A	100	85
B	150	40
C	2000	150

TARA

KODAS	MIN	MAX
P1	50	100
P2	100	200
P3	0	50
P4	100	150

Užklausa: Išrinkti gaminiui tarą pagal svorį.

Atsakymas:

GAMINYS	SVORIS	TARA
A	85	P1
B	40	P3
C	150	P2
C	150	P4

$$(\pi_{\text{KODAS, SVORIS}}(\text{GAMINIAI})) \mid_{><} (\pi_{\text{KODAS}}(\text{TARA}))$$

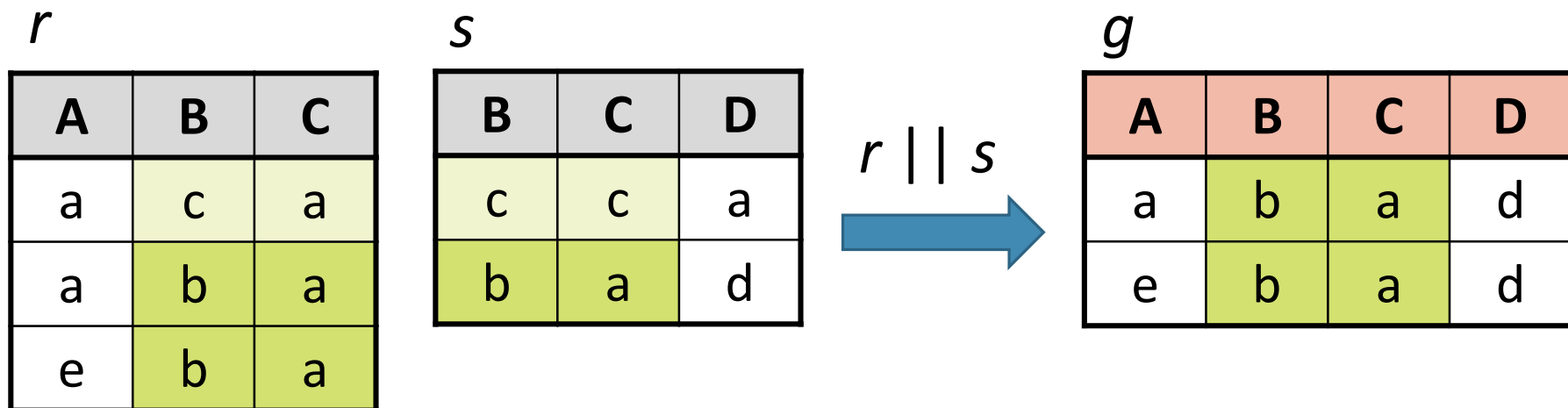
$$(\text{SVORIS} > \text{MIN}) \wedge (\text{SVORIS} \leq \text{MAX})$$

```
SELECT gaminiai.kodas AS gaminys, svoris, tara.kodas
AS tara
FROM gaminiai, tara
WHERE svoris > min AND svoris <= max ;
```

RA. Natūralioji jungtis (angl. *Natural Joins*)

Natūraliosios jungties operacija santykiams r ir s taikoma tada, kai šių santykių schemas R ir S turi bendrų atributų $A_i \in R \cap S$. Atlikus operaciją, gaunamas naujos schemas G santykis g . Operacija žymima $g = r \parallel s$. Ji realizuojama nuosekliai atliekant dekartinės sandaugos, selekcijos ir projekcijos operacijas:

- Skaičiuojama sandauga $r \times s$;
- Atrenkami tie kortežai, kurių A_i komponentai iš r ir s tuose pačiuose kortežuose sutampa;
- Pašalinami pasikartojantys stulpeliai $A_i \in S$ (arba $A_i \in R$).



Natūralios jungties realizacija SQL

KOMPL

NR	PAVAD	KAINA
001038	Klevas	150
001027	Liepa	50
002003	Uosis	1000

SAND

NR	KIEKIS	VIETA	AR_PAPILDYTI
001038	100	Kaunas	.F.
001027	500	Kaunas	.F.
001027	75	Vilnius	.F.
002003	5	Alytus	.T.

Užklausa: Pateikite baldų komplektų, kurių nereikia papildyti sandėliuose, numerius ir pavadinimus.

$$(\pi_{NR, PAVAD} (KOMPL)) \mid >< \mid (\pi_{NR} (\sigma_{VIETA='Kaunas' \wedge AR_PAPILDYTI=.F.} (SAND)))$$

Atsakymas:

NR	PAVAD
001038	Klevas
001027	Liepa

```
SELECT kompl.nr, pavad
FROM kompl, sand
WHERE vieta = 'Kaunas' AND (NOT
ar_papildyti) AND kompl.nr = sand.nr ;
```

PIRMOS DALIES PABAIGA
