

[http://www.dcs.fmph.uniba.sk/~plachetk
/TEACHING/DB1](http://www.dcs.fmph.uniba.sk/~plachetk/TEACHING/DB1)

Tomáš Plachetka

Fakulta matematiky, fyziky a informatiky,
Univerzita Komenského, Bratislava

Zima 2025–2026

- **Databáza je štruktúra** pre relačný (predikátový) kalkul
- **Dotaz je formula** $\varphi(X_1, \dots, X_n)$, kde $X_1, \dots, X_n$ sú voľné premenné
- **Výsledok dotazu** $\varphi(X_1, \dots, X_n)$ je množina usporiadaných n-tíc $[X_1, \dots, X_n]$, pre ktoré platí $\varphi(X_1, \dots, X_n)$

Ako **počítať** výsledok dotazu? O tom je **relačná algebra**.

Operandami relačnej algebry sú relácie. Relácia sa dá reprezentovať tabuľkou, kde mená atribútov označujú stĺpce tabuľky (hlavičkový riadok). Zatiaľ predpokladajme, že tabuľka je **množina** riadkov (bez duplikátov)

V ďalšom texte bude **X** označovať vektor atribútov $[X_1, \dots, X_n]$, **Y** bude označovať vektor atribútov $[Y_1, \dots, Y_n]$ a podobne

- **Formálna špecifikácia SQL.** Hoci syntax SELECT je veľmi „barokná“, sémantika sa dá presne popísať relačnou algebrou
- **Optimalizácia dotazov** v DBMS. SELECT sa dá priamočiaro transformovať na ekvivalentný výraz relačnej algebry (operátorový strom)
 - Pri optimalizácii sa tento výraz transformuje na ekvivalentný, výpočtovo efektívnejší algebraický výraz (resp. na postupnosť priradení), ktorému sa hovorí **logický plán**
 - Logický plán zapísaný v relačnej algebre sa namapuje do **fyzických operátorov** konkrétneho DBMS. Takto vznikne **fyzický plán** výpočtu dotazu

Pohľad programátora:

- Dotaz, resp. program **v relačnom kalkule, Datalogu a SQL** vyjadruje **ČO** treba vypočítať (**nie ako**). Nie je príliš dôležité, ako “efektívne” je program napísaný. O spôsob výpočtu (a optimalizáciu) sa stará stroj, nie programátor
- Dotaz, resp. program **v relačnej algebre** vyjadruje **AKO** sa vypočíta výsledná relácia z EDB relácií

(V konečnom dôsledku z AKO vyplýva aj ČO sa počíta. Lenže ľudský spôsob myslenia je „najskôr ČO, až potom AKO“, t.j. „najskôr špecifikácia, až potom implementácia“)

- **Zjednotenie, prienik, rozdiel** (vyžaduje sa, aby relácie mali rovnakú schému, t.j. aby boli rovnakého typu)
- **Selekcia**: výber riadkov
- **Projekcia**: výber stĺpcov
- **Kartézsky súčin a join**: skladanie relácií
- **Premenovanie** relácií a atribútov relácií
- ...

Relačná algebra

$$r_1 \cup r_2 =$$

$$r_1 \cap r_2 =$$

$$r_1 - r_2 =$$

Relačný kalkul:

$$\{\mathbf{X}: r_1(\mathbf{X}) \vee r_2(\mathbf{X})\}$$

$$\{\mathbf{X}: r_1(\mathbf{X}) \wedge r_2(\mathbf{X})\}$$

$$\{\mathbf{X}: r_1(\mathbf{X}) \wedge \neg r_2(\mathbf{X})\}$$

Negácia sa vyjadruje ako rozdiel relácií (inak sa ani vyjadriť nedá). Relácia r_1 vystupuje ako „pozitívny kontext“, t.j. obsahuje množinu kandidátov na výsledok. Negácia spôsobí len vynechanie niektorých n-tíc z r_1 (tých, ktoré sú v r_2)

Projekcia: π

Kalkul:

Nech r je typu $r(\mathbf{X}, \mathbf{Y})$. Potom

$$\pi_{\mathbf{X}}(r) = \{\mathbf{X}: \exists \mathbf{Y} r(\mathbf{X}, \mathbf{Y})\}$$

Procedurálna definícia: $r_2 := \pi_{\mathbf{X}}(r_1)$

- r_2 vzniká kopírovaním riadkov r_1 , pričom pre každý riadok r_1 sa do r_2 skopírujú len tie atribúty, ktoré sú v $\mathbf{X}$
- Nakoniec treba eliminovať duplikované riadky v r_2 (ak počítame s množinami)

Príklad (Ullman)

Relation sells

Bar	Beer	Price
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Miller	3.00

$\text{prices} := \pi_{\text{Beer, Price}}(\text{sells})$

Beer	Price
Bud	2.50
Miller	2.75
Miller	3.00

Kalkul:

- Nech r je typu $r(\mathbf{X})$. Potom

$$\sigma_{c(\mathbf{X})}(r) = \{\mathbf{X}: r(\mathbf{X}) \wedge c(\mathbf{X})\}$$

Procedurálna definícia: $r_2 := \sigma_c(r_1)$

- r_2 vzniká kopírovaním riadkov r_1 , pričom do r_2 sa skopírujú len tie riadky, pre ktoré platí podmienka c

Príklad (Ullman)

Relation sells

Bar	Beer	Price
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Miller	3.00

joe_menu := $\sigma_{\text{Bar}=\text{"Joe's"}}(\text{sells})$

Bar	Beer	Price
Joe's	Bud	2.50
Joe's	Miller	2.75

Kartézsky súčin: $\times$

Kalkul:

- Nech r_1 je typu $r_1(\mathbf{X})$ a r_2 je typu $r_2(\mathbf{Y})$, pričom $\mathbf{X} \cap \mathbf{Y} = \emptyset$. Potom $r_1 \times r_2 = \{[\mathbf{X}, \mathbf{Y}]: r_1(\mathbf{X}) \wedge r_2(\mathbf{Y})\}$

Procedurálna definícia: $r_3 := r_1 \times r_2$

- r_3 vzniká kopírovaním všetkých dvojíc riadkov r_1 a r_2
- Spoločné atribúty v r_1 a r_2 jednoducho nedovolíme, aby sme sa vyhli problému v pomenovaní atribútov r_3 . (Vlastne môžeme dovoliť aj spoločné atribúty, ale vo výsledku potom musíme dôsledne používať prefixy atribútov—aby bolo jasné, z ktorej relácie pochádzajú.)

Príklad (Ullman)

$r3 := r1 \times r2$

r1(

A,	B
1	2
3	4

)

r2(

B,	C
5	6
7	8
9	10

)

r3(

A,	r1.B,	r2.B,	C
1	2	5	6
1	2	7	8
1	2	9	10
3	4	5	6
3	4	7	8
3	4	9	10

)

Join (theta-join): $\bowtie_c$

Kalkul:

- Nech r_1 je typu $r_1(\mathbf{X})$ a r_2 je typu $r_2(\mathbf{Y})$, pričom $\mathbf{X} \cap \mathbf{Y} = \emptyset$. Potom

$$r_1 \bowtie_{c(\mathbf{X}, \mathbf{Y})} r_2 = \{[\mathbf{X}, \mathbf{Y}]: r_1(\mathbf{X}) \wedge r_2(\mathbf{Y}) \wedge c(\mathbf{X}, \mathbf{Y})\}$$

Procedurálna definícia: $r_3 := r_1 \bowtie_{c(\mathbf{X}, \mathbf{Y})} r_2$

- r_3 vzniká kopírovaním všetkých dvojíc riadkov r_1 a r_2 , pričom do r_3 sa skopírujú len tie dvojice, pre ktoré platí podmienka c
- Spoločné atribúty v r_1 a r_2 jednoducho nedovolíme, aby sme sa vyhli problému v pomenovaní atribútov r_3 . (Vlastne môžeme dovoliť aj spoločné atribúty, ale vo výsledku potom musíme dôsledne používať mená relácií ako prefixy atribútov.)
- **Theta-join je selekcia aplikovaná na kartézsky súčin**

Príklad (Ullman)

sells(

Bar,	Beer,	Price
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Coors	3.00

)

bars(

Name,	Addr
Joe's	Maple St.
Sue's	River Rd.

)

bar_info := sells $\bowtie_{\text{sells.bar} = \text{bars.name}}$ **bars**

bar_info(

Bar,	Beer,	Price,	Name,	Addr
Joe's	Bud	2.50	Joe's	Maple St.
Joe's	Miller	2.75	Joe's	Maple St.
Sue's	Bud	2.50	Sue's	River Rd.
Sue's	Coors	3.00	Sue's	River Rd.

)

Premenovanie: ρ

Kalkul:

- Premenovanie je bežná substitúcia používaná matematikmi. Substituovať možno nielen mená atribútov, ale aj meno relácie

Procedurálna definícia: $r_2 := \rho_{r_2(\gamma)}(r_1)$

- r_2 je kópiou r_1 , len niektoré jej atribúty sa volajú inak

Konkrétna syntax operátora ρ nie je príliš dôležitá. Avšak musí z nej byť jasné, ktoré meno je pôvodné a ktoré meno je nové.

Napríklad, pre reláciu $\text{lubi}(\text{Pijan}, \text{Alkohol})$ sa $\rho_{\text{Ochmelka} := \text{Pijan}}(\text{lubi})$ chápe rovnako ako $\rho_{\text{Ochmelka} \leftarrow \text{Pijan}}(\text{lubi})$ alebo $\rho_{\text{Ochmelka}, \text{Alkohol}}(\text{lubi})$

Príklad (Ullman)

bars(

Name,	Addr
Joe's	Maple St.
Sue's	River Rd.

)

$\rho_{r(\text{Bar}, \text{Addr})}$ bars

r(

Bar,	Addr
Joe's	Maple St.
Sue's	River Rd.

)

Natural join: $\bowtie$

Kalkul:

- Nech r_1 je typu $r_1(\mathbf{X}, \mathbf{Z})$ a r_2 je typu $r_2(\mathbf{Y}, \mathbf{Z})$, pričom $\mathbf{Z}$ sú spoločné atribúty r_1 a r_2 . Potom

$$r_1 \bowtie r_2 = \{[\mathbf{X}, \mathbf{Y}, \mathbf{Z}]: r_1(\mathbf{X}, \mathbf{Z}) \wedge r_2(\mathbf{Y}, \mathbf{Z})\}$$

Procedurálna definícia: $r_3 := r_1 \bowtie r_2$

- r_3 vzniká kopírovaním všetkých dvojíc riadkov r_1 a r_2 , pričom spoločné atribúty (atribúty s rovnakým menom) sú testované na rovnosť a sú kopírované iba raz
- Natural join sa dá vyjadriť pomocou premenovania, theta-joinu a projekcie. Často je však prirodzené **zlúčiť dve relácie do jednej na základe rovnosti spoločných atribútov**

Natural join: $\bowtie$

Príklad (Ullman)

sells(	Bar,	Beer,	Price	)
	Joe's	Bud	2.50	
	Joe's	Miller	2.75	
	Sue's	Bud	2.50	
	Sue's	Coors	3.00	

bars(	Name,	Addr	)
	Joe's	Maple St.	
	Sue's	River Rd.	

$\text{bar_info} := \text{sells} \bowtie \rho_{\text{bars}(\text{Bar}, \text{Addr})}(\text{bars})$

bars.Name bolo treba premenovať na **bars.Bar**

bar_info(	Bar,	Beer,	Price,	Addr	)
	Joe's	Bud	2.50	Maple St.	
	Joe's	Miller	2.75	Maple St.	
	Sue's	Bud	2.50	River Rd.	
	Sue's	Coors	3.00	River Rd.	

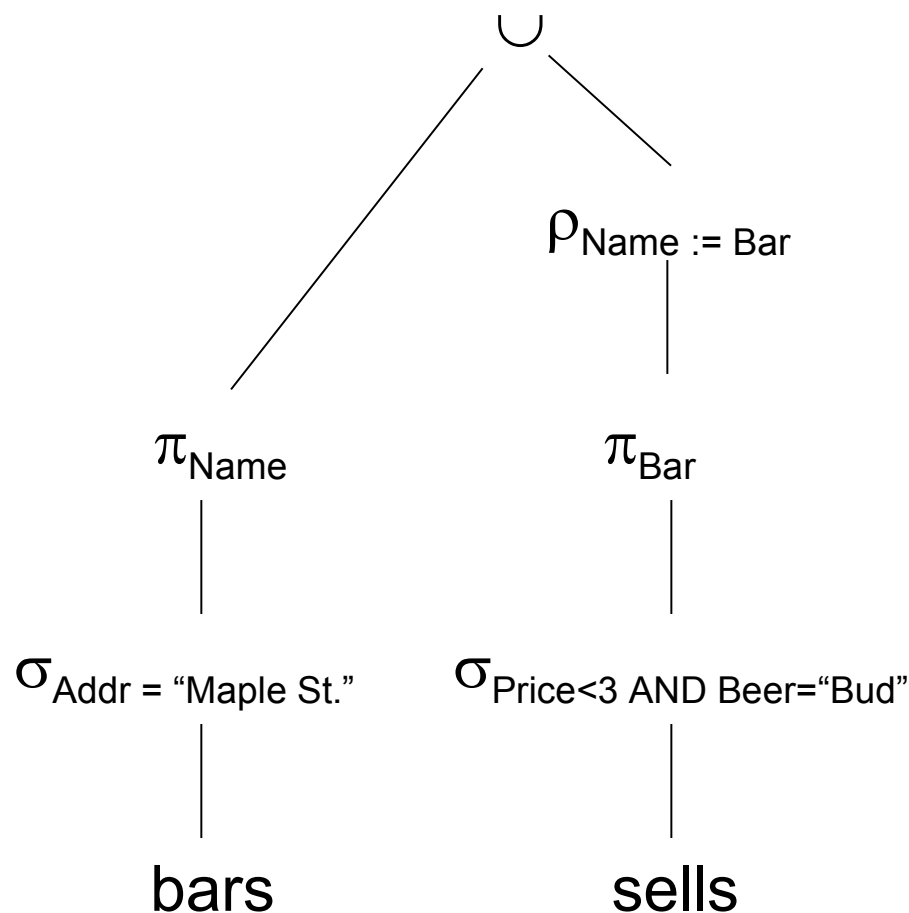
Tri formy notácie výrazu: bary, ktoré bud' sídli na Maple St., alebo predávajú Bud za menej ako 3\$ (Ullman)

Strom výrazu

Výraz s operátormi

$$\pi_{\text{Name}}(\sigma_{\text{Addr} = \text{"Maple St."}}(\text{bars})) \cup \rho_{\text{Name} := \text{Bar}}(\pi_{\text{Bar}}(\sigma_{\text{Price} < 3 \text{ AND Beer} = \text{"Bud"}}(\text{sells})))$$

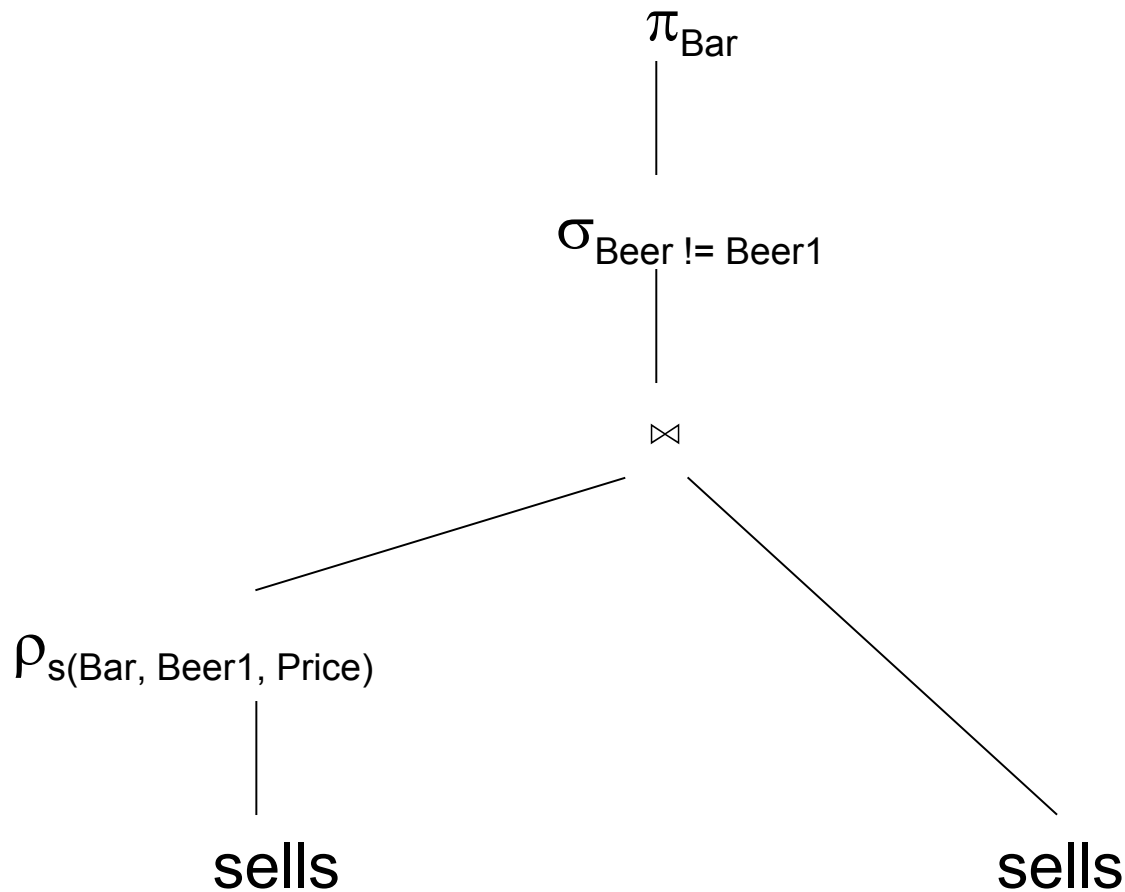
Postupnosť priradení

$$r := \pi_{\text{Name}}(\sigma_{\text{Addr} = \text{"Maple St."}}(\text{bars});$$
$$s := \rho_{\text{Name} := \text{Bar}}(\pi_{\text{Bar}}(\sigma_{\text{Price} < 3 \text{ AND Beer} = \text{"Bud"}}(\text{sells})));$$
$$t := r \cup s$$


Ekvivalentné zápisy výrazov relačnej algebry

Ešte príklad (Ullman): bary, ktoré predávajú (aspoň) dva rôzne druhy piva za rovnakú cenu

$\pi_{\text{Bar}}(\sigma_{\text{Beer} \neq \text{Beer1}}(\rho_{\text{s(Bar, Beer1, Price)}}(\text{sells}) \bowtie \text{sells}))$



Niektoré zákony (množinovej) relačnej algebry

- Prirodzené spojenie (natural join) a zjednotenie sú komutatívne, asociatívne a idempotentné

- Platia distributívne zákony

$$r \bowtie (s \cup t) = (r \bowtie s) \cup (r \bowtie t)$$

$$r \bowtie (s - t) = (r \bowtie s) - (r \bowtie t)$$

- Ak $Y \subseteq X$, tak potom $\pi_Y \pi_X(r) = \pi_Y(r)$

- Ak podmienka c neobsahuje atribúty s , tak potom $\sigma_c(r \times s) = \sigma_c(r) \times s$

- Ak podmienka c_{rs} obsahuje atribúty r aj s , podmienka c_r obsahuje len atribúty r , a podmienka c_s obsahuje len atribúty s , tak potom

$$\sigma_{c_{rs} \wedge c_r \wedge c_s}(r \times s) = \sigma_{c_r}(r) \bowtie_{c_{rs}} \sigma_{c_s}(s)$$

Optimalizácia na úrovni relačnej algebry: príklad

dodava

Firma	Vyrobok	Cena	Lehota
-------	---------	------	--------

firmy

Firma	Mesto
-------	-------

objednavky

Klient	Vyrobok
--------	---------

Ktorý klient objednal výrobok, ktorý vie dodať niektorá firma z Nuernberg?

SELECT o.Klient

FROM objednavky o, dodava d, firmy f

WHERE f.Mesto = 'nuernberg' **and** f.Firma = d.Firma
and d.Vyrobok = o.Vyrobok

Optimalizácia na úrovni relačnej algebry: príklad

Ktorý klient objednal výrobok, ktorý vie dodať niektorá firma z Nuernberg?

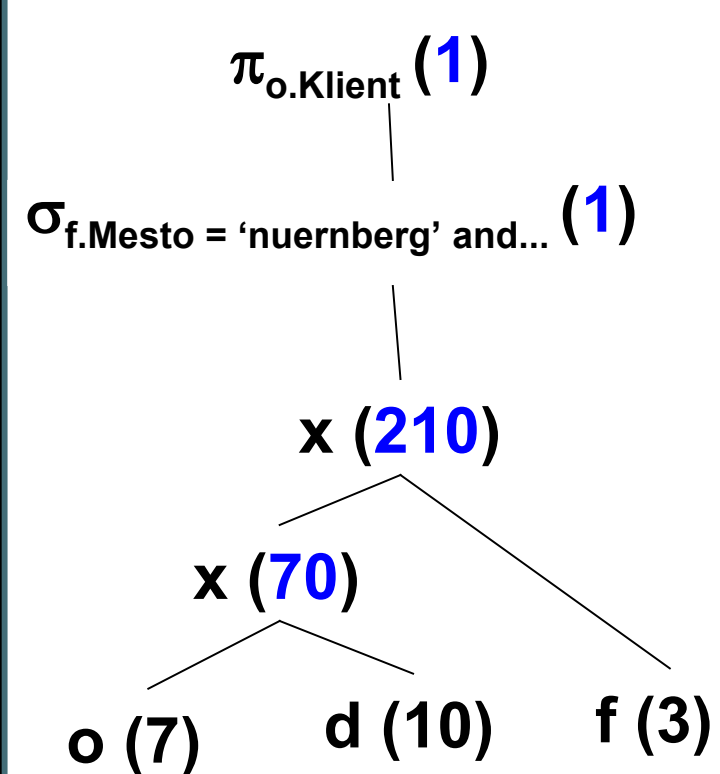
<i>dodava</i>			
Firma	Vyrobok	Cena	Lehota
vobis	pc386	2000	4
quelle	pc386	1900	9
vobis	pc486	2900	4
escom	pc486	3000	5
vobis	pc586	5000	7
escom	pc586	5900	9
vobis	hp4l	1400	6
vobis	hddisk	13	0
escom	hddisk	12	0
quelle	cdrom	400	4

<i>dodavatelia</i>	
Firma	Mesto
vobis	ulm
escom	ulm
quelle	nuernberg

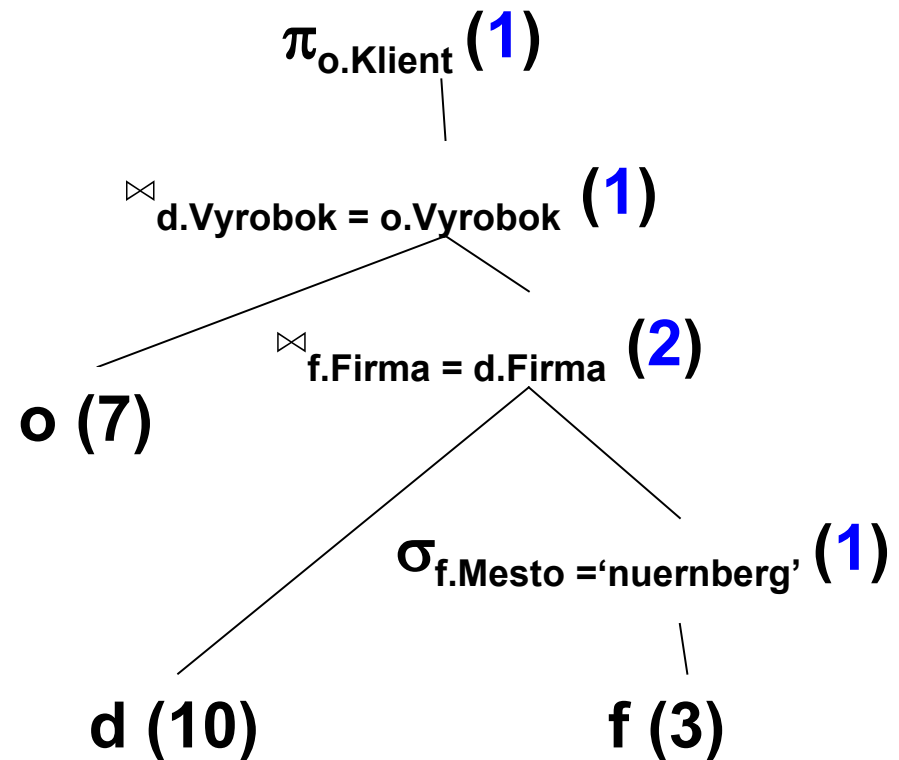
<i>objednavky</i>	
Klient	Vyrobok
meier	pc486
meier	hddisk
reich	pc586
reich	hp4l
reich	hddisk
arm	pc386
arm	hddisk

Optimalizácia na úrovni relačnej algebry: príklad

SELECT o.Klient
FROM objednávky o, dodava d, firmy f
WHERE f.Mesto = 'nuernberg' **and** f.Firma = d.Firma
and d.Vyrobok = o.Vyrobok



Medzivýsledok: **282 riadkov**



Medzivýsledok: **5 riadkov**

Výpočet jednoduchého SELECT

SELECT $A_1, \dots, A_n$
FROM $r_1, \dots, r_m$
WHERE c

Projekcia selekcie kartézského súčinu

$$\pi_{A_1, \dots, A_n} \sigma_c(r_1 \times \dots \times r_m)$$

Projekcia joinu

$$\pi_{A_1, \dots, A_n} (r_1 \bowtie_c \dots \bowtie_c r_m)$$

Toto zhruba vyjadruje „kanonický“ výpočet výsledku príkazu SELECT (ktorý sa môže ďalej optimalizovať). Zatiaľ sme neriešili napr. spracovanie **duplikátov**

Multimnožiny (bags)

- **Multimnožina (bag)** je množina s duplikátmi. Napríklad {1, 2, 3, 1, 2} je multimnožina. Aj {1, 2, 3} je multimnožina. Každá množina je multimnožinou, ale nie nutne naopak
- **SQL počíta nad multimnožinami.** Dôvodom je snaha ušetriť, eliminácia duplikátov je rovnako zložitá ako triedenie. (Na druhej strane, skutočne sa šetrí, keď je duplikátov veľa?)
- **Relačná algebra počíta nad multimnožinami**
- Operátory relačnej algebry sa dajú definovať aj pre multimnožiny
 - *Zjednotenie, prienik a rozdiel* treba poopraviť (pre UNION **ALL**, ...)
 - *Projekcia* pre multimnožiny neeliminuje duplikáty
 - *Selekcia, kartézsky súčin a joiny* sú definované ako predtým (podľa tabuľkovej sémantiky)

Príklad (Ullman)

$r($

A,	B
1	2
5	6
1	2

)

$\sigma_{A < 3 \wedge B < 4}(r) =$

A	B
1	2
1	2

Príklad (Ullman)

$r($

A,	B
1	2
5	6
1	3

)

$\pi_A(r) =$

A
1
5
1

Multimnožiny (bags)

Príklad (Ullman)

$r($

A,	B
1	2
5	6
1	2

)

$s($

B,	C
3	4
7	8

)

$r \times s = A$

r.B	s.B	C	
1	2	3	4
1	2	7	8
5	6	3	4
5	6	7	8
1	2	3	4
1	2	7	8

Multimnožiny (bags)

Príklad (Ullman)

$r($

A,	B
1	2
5	6
1	2

)

$s($

B,	C
3	4
7	8

)

$r \bowtie_{r.B < s.B} s =$

A	r.B	s.B	C
1	2	3	4
1	2	7	8
5	6	7	8
1	2	3	4
1	2	7	8

Multimnožiny (bags)

Zjednotenie: multimnožiny sa „zreťazia“.

Príklad: $\{1, 2, 1\} \cup \{1, 1, 2, 3, 1\} = \{1, 1, 1, 1, 1, 2, 2, 3\}$

Prienik: vo výsledku sa prvok objaví toľkokrát, koľkokrát je minimum jeho výskytu v operandoch

Príklad: $\{1, 2, 1, 1\} \cap \{1, 2, 1, 3\} = \{1, 1, 2\}$

Rozdiel: vo výsledku sa prvok objaví toľkokrát, koľkokrát sa vyskytuje v prvom operande mínus koľkokrát sa vyskytuje v druhom operande (samozrejme, aspoň nulakrát)

Príklad: $\{1, 2, 1, 1\} - \{1, 2, 3\} = \{1, 1\}$

Multimnožiny (bags)

Pozor, **nie všetky vlastnosti operácií s množinami sú zachované pre multimnožiny!**

Príklad (Ullman): zjednotenie množín je idempotentné (t.j. $s \cup s = s$), ale zjednotenie multimnožín nie je

Ďalšie operátory relačnej algebry

- Eliminácia duplikátov: δ
- (Triedenie: T)
- OUTERJOIN
- Grupovanie a agregácia: Γ

- $r_2 := \delta(r_1)$

r_2 je kópiou r_1 , ale bez duplikovaných riadkov

Príklad (Ullman)

$r = ($

A	B
1	2
3	4
1	2

$)$

$\delta(r) =$

A	B
1	2
3	4

- $T_{x_1, \dots, x_n}(r)$

je **zoznam**, ktorý vznikol utriedením r najprv podľa X_1 , potom podľa X_2 , ..., nakoniec podľa X_N (rovnosti sa riešia náhodne). Ak je pred niektorým atribútom šípka nadol, tak sa triedi zostupne, inak sa triedi vzostupne

Príklad (Ullman)

$r = ($

A	B
1	2
3	4
5	2

$)$

$$T_B(r) = [(5, 2), (1, 2), (3, 4)]$$

Full join: OUTERJOIN

- $r_3 := r_1 \text{ OUTERJOIN } r_2$

Full join je podobný theta-joinu, ale do výsledku navyše pribudnú riadky z r_1 a r_2 , ktoré sa s ničím nespájajú. Chýbajúce hodnoty v týchto riadkoch sa doplnia špeciálnymi hodnotami **null**

Príklad (Ullman)

$$r_1 = \left(\begin{array}{|c|c|} \hline A & B \\ \hline 1 & 2 \\ 4 & 5 \\ \hline \end{array} \right)$$
$$r_2 = \left(\begin{array}{|c|c|} \hline B & C \\ \hline 2 & 3 \\ 6 & 7 \\ \hline \end{array} \right)$$

[1, 2] sa v $r_1 \bowtie r_2$ spája s [2, 3], ale [4, 5] a [6, 7] sa nespájajú s ničím

$$r_1 \text{ OUTERJOIN } r_2 = \begin{array}{|c|c|c|} \hline A & B & C \\ \hline 1 & 2 & 3 \\ 4 & 5 & \text{null} \\ \text{null} & 6 & 7 \\ \hline \end{array}$$

$$r_2 := \Gamma_x(r_1)$$

Vo vektore **X** môžu byť použité

- **grupovacie atribúty** (atribúty z r_1)
- **agregácie** tvaru AGG(Y), kde Y je atribútom r_1 (tzv. agregovaný atribút) a AGG je niektorá z agregačných funkcií SUM, COUNT, AVG, STDEV, MAX, MIN

Operátor $\Gamma_x(r_1)$

1. vyrobí z relácie r_1 skupiny, pričom riadky v každej zo skupín sa zhodujú vo všetkých hodnotách grupovacích atribútov
2. vypočíta všetky agregácie, pre každú skupinu zvlášť

Výsledkom je tabuľka, v ktorej každej skupine prináleží jeden riadok

Príklad (Ullman) $r =$ (

A	B	C	D
1	2	3	1
4	5	6	2
1	2	5	3

)

$$\Gamma_{A, B, \text{AVG}(C)}(r) = ??$$

1.krok: zgrupuj r podľa A a B :

A	B	C	D
1	2	3	1
1	2	5	2
4	5	6	3

2.krok: vypočítaj AVG(C)
pre každú grupu:

A	B	AVG(C)
1	2	4
4	5	6

$\Gamma_{A_1, A_2, \dots, A_n}(r)$ je agregáčny operátor aplikovaný na r :

- A_i je buď **grupovací atribút** alebo **agregácia**

[SUM(A_i) | COUNT(A_i) | AVG(A_i) | STDEV(A_i) | MIN(A_i), MAX(A_i)]

(kde A_i je **agregovaný atribút**). Nech **G** je množina grupovacích atribútov a **A** je množina agregovaných atribútov. (r môže obsahovať aj iné atribúty, t.j. také, ktoré nepatria do **G** ani do **A**.)

- Medzivýsledkom je relácia r' , ktorá vzniká projekciou r na množinu atribútov **G** s následným odstránením duplikátov:
 $r' := \delta(\Pi_G(r))$. **Jeden riadok r' zodpovedá skupine riadkov v r**
- Výsledkom je relácia r' rozšírená o výsledky agregácií, pričom hodnota agregácie v riadku výsledku sa vypočíta aplikáciou agregáčnej funkcie na hodnoty agregovaného atribútu v zodpovedajúcej skupine riadkov v r

Syntax a sémantika **SELECT** s **GROUP BY** a **HAVING**:

SELECT <S_attr> **5**

FROM $r_1, r_2, \dots, r_n$ **1**

WHERE <w_cond> **2**

GROUP BY <G_attr> **3**

HAVING <h_cond> **4**

$$\pi_{S_attr}(\sigma_{h_cond}(\Gamma_{G_attr, AGG(Attr)}(\sigma_{w_cond}(r_1 \times r_2 \times \dots \times r_n))))$$

5 **4** **3** **2** **1**