



BAZE PODATAKA

Upitni Jezici
(Query Languages)



Upitni Jezici

- Relaciona Algebra
- Relacioni Račun
- SQL



Upitni jezici

- Jezici koji omogućuju dobijanje informacija iz baze.
- Kategorije upitnih jezika
 - Proceduralni
 - Ne-proceduralni, ili deklarativni
- “Čisti” jezici:
 - Relaciona algebra
 - Relacioni račun sa torkama (Tuple relational calculus)
 - Relacioni račun sa domenima (Domain relational calculus)
- “Čisti” jezici predstavljaju osnovu komercijalnih upitnih jezika.



Relaciona Algebra

- Proceduralni jezik
- Šest osnovnih operacija
 - Selekcija (select): σ
 - Projekcija (project): Π
 - Unija (union): \cup
 - Razlika (set difference): $-$
 - Dekartov Proizvod (Cartesian product): \times
 - Preimenovanje (rename): ρ
- Operacije su definisane na jednoj (σ, Π) ili dve relacije ($\cup, -, \times$) i kao rezultat daju novu relaciju.



Selekcija (primer)

Relacija r

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

$\sigma_{A=B \wedge D > 5}(r)$

A	B	C	D
α	α	1	7
β	β	23	10



Operacija Selekcije

- Notacija: $\sigma_p(r)$
- p je predikat selekcije
- Definicija:

$$\sigma_p(r) = \{t \mid t \in r \text{ and } p(t)\}$$

gde je p formula predikatnog računa koja se sastoji od članova povezanih sa : \wedge (**i**), \vee (**ili**), \neg (**ne**)
Svaki član je:

<atribut> op <atribut> ili <konstanta>

gde je op : =, \neq , >, \geq , <, \leq

- Primer:

$$\sigma_{branch_name="Perryridge"}(account)$$



Projekcija (primer)

Relation r :

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

$\Pi_{A,C}(r)$

A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2



Operacija Projekcije

- Notacija: $\Pi_{A_1, A_2, \dots, A_k}(r)$

gde su A_1, A_2 atributi a r relacija.

- Rezultat je relacija od k kolona koje su navedene kao predikati operacije
- Vrste koje se ponavljaju se uklanjaju iz rezultata, jer su relacije skupovi
- Primer: Eliminisanje atributa *branch_name* iz *account*

$$\Pi_{account_number, balance}(account)$$



Unija (primer)

Relacije r , s :

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

$r \cup s$:

A	B
α	1
α	2
β	1
β	3



Operacija Unije

- Notacija: $r \cup s$
- Definicija:
 - $r \cup s = \{t \mid t \in r \text{ ili } t \in s\}$
- Da bi operacija $r \cup s$ bila validna:
 1. r, s moraju imati isti **broj** atributa
 2. Domeni atributa moraju biti **kompatibilni**
- Primer: naći sve klijente koji imaju bilo račun bilo kredit

$\Pi_{customer_name}(depositor) \cup \Pi_{customer_name}(borrower)$



Razlika (primer)

Relacije r , s :

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

$r - s$:

A	B
α	1
β	1



Operacija Razlike

- Notacija: $r - s$

- Definicija:

$$r - s = \{t \mid t \in r \text{ i } t \notin s\}$$

- Razlika je moguća između **kompatibilnih** relacija.

- r i s moraju imati isti broj atributa

- Domeni atributa r i s moraju biti kompatibilni



Dekartov Proizvod (primer)

Relacije **r**, **s**:

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

r X s:

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b



Operacija Dekartov Proizvod

- Notacija $r \times s$

- Definicija:

$$r \times s = \{tq \mid t \in r \text{ i } q \in s\}$$

- Pretpostavlja se da su skupovi atributi $r(R)$ i $s(S)$ uzajamno isključivi. ($R \cap S = \emptyset$).
- Ako skupovi atributa $r(R)$ i $s(S)$ nisu isključivi, moraju se preimenovati.



Kombinovanje operacija (primer)

- Mogu se praviti izrazi korišćenjem više operacija

- Primer: $\sigma_{A=C}(r \times s)$

- $r \times s$

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

- $\sigma_{A=C}(r \times s)$

A	B	C	D	E
α	1	α	10	a
β	2	β	10	a
β	2	β	20	b



Operacija Preimenovanja

- Omogućuje imenovanje, i referisanje, rezultata izraza u relacionoj algebri.
- Omogućuje referisanje na relaciju sa više imena.
- Primer:

$$\rho_X(E)$$

vraća izraz E pod imenom X

- Ako izraz relacione algebre E ima n atributa, tada

$$\rho_{X(A_1, A_2, \dots, A_n)}(E)$$

vraća rezultat izraza E pod imenom X , sa atributima preimenovanim u A_1, A_2, \dots, A_n .



Banka (primer)

branch (*branch_name*, *branch_city*, *assets*)

customer (*customer_name*, *customer_street*,
customer_city)

account (*account_number*, *branch_name*, *balance*)

loan (*loan_number*, *branch_name*, *amount*)

depositor (*customer_name*, *account_number*)

borrower (*customer_name*, *loan_number*)



Primeri (1)

- Find all loans of over \$1200

$\sigma_{amount > 1200} (loan)$

- Find the loan number for each loan of an amount greater than \$1200

$\Pi_{loan_number} (\sigma_{amount > 1200} (loan))$



Primeri (2)

- Find the names of all customers who have a loan, an account, or both, from the bank

$$\Pi_{customer_name}(borrower) \cup \Pi_{customer_name}(depositor)$$

- Find the names of all customers who have a loan and an account at bank.

$$\Pi_{customer_name}(borrower) \cap \Pi_{customer_name}(depositor)$$



Primeri (3)

- Find the names of all customers who have a loan at the Perryridge branch.

$$\Pi_{customer_name} (\sigma_{branch_name="Perryridge"} (\sigma_{borrower.loan_number = loan.loan_number}(borrower \times loan)))$$

- Find the names of all customers who have a loan at the Perryridge branch but do not have an account at any branch of the bank.

$$\Pi_{customer_name} (\sigma_{branch_name = "Perryridge"}$$
$$(\sigma_{borrower.loan_number = loan.loan_number}(borrower \times loan))) - \Pi_{customer_name}(\text{depositor})$$



Primeri (4)

- Find the names of all customers who have a loan at the Perryridge branch.

- Query 1

$$\Pi_{\text{customer_name}} (\sigma_{\text{branch_name} = \text{"Perryridge"}} (\sigma_{\text{borrower.loan_number} = \text{loan.loan_number}} (\text{borrower} \times \text{loan})))$$

- Query 2

$$\Pi_{\text{customer_name}} (\sigma_{\text{loan.loan_number} = \text{borrower.loan_number}} (\sigma_{\text{branch_name} = \text{"Perryridge"}} (\text{loan})) \times \text{borrower}))$$



Primeri (5)

- Find the largest account balance
 - Strategy:
 - Find those balances that are *not* the largest
 - Rename *account* relation as *d* so that we can compare each account balance with all others
 - Use set difference to find those account balances that were *not* found in the earlier step.
 - The query is:

$$\Pi_{balance}(account) - \Pi_{account.balance}$$

$$(\sigma_{account.balance < d.balance} (account \times \rho_d (account)))$$



Formalna Definicija Relacione Algebre

- Izraz relacione algebre sastoji se od:
 - Relacije u bazi podataka
 - Konstantne relacije
- Neka su E_1 i E_2 izrazi relacione algebre; Tada su izrazi relacione algebre i:
 - $E_1 \cup E_2$
 - $E_1 - E_2$
 - $E_1 \times E_2$
 - $\sigma_p(E_1)$, P je predikat na atributima iz E_1
 - $\Pi_S(E_1)$, S je lista nekih atributa iz E_1
 - $\rho_x(E_1)$, x je novo ime za rezultat E_1



Dodatne Operacije

- Definišemo dodatne operacije koje pojednostavljaju uobičajene upite.
 - **Presek** (Set intersection)
 - **Prirodno Spajanje** (Natural join)
 - **Deljenje** (Division)
 - **Dodeljivanje** (Assignment)



Presek

- Notacija: $r \cap s$

- Definicija:

$$r \cap s = \{ t \mid t \in r \text{ i } t \in s \}$$

- Pretpostavka:

- r, s imaju isti broj atributa

- atributi relacija r i s su kompatibilni

- Definicija osnovnim operacijama relacione algebre:

$$r \cap s = r - (r - s)$$



Presek (primer)

Relacije r , s :

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

$r \cap s$

A	B
α	2



Prirodno Spajanje

- Notacija: $r \bowtie s$
- Neka su r i s relacije na šemama R i S respektivno. Tada je, $r \bowtie s$ relacija na šemi $R \cup S$ koja se dobija na sledeći način:
 - Posmatrajmo neki par torki t_r iz r i t_s iz s .
 - Ako t_r i t_s imaju istu vrednost na atributima u $R \cap S$, rezultatu treba dodati torku t , takvu da
 - t ima istu vrednost kao t_r na r
 - t ima istu vrednost kao t_s na s
- Primer: $R = (A, B, C, D)$, $S = (E, B, D)$
 - Šema rezultata = (A, B, C, D, E)
 - $r \bowtie s$ je definisano kao:

$$\Pi_{r.A, r.B, r.C, r.D, s.E}(\sigma_{r.B = s.B \wedge r.D = s.D}(r \times s))$$



Prirodno Spajanje (primer)

Relacije r , s :

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ϵ

s

$r \bowtie s$

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ



Deljenje

- Notacija: $r \div s$
- Za upite koji uključuju frazu “za sve”.
- Neka su r i s relacije na šemama R i S respektivno takve da je
 - $R = (A_1, \dots, A_m, B_1, \dots, B_n)$
 - $S = (B_1, \dots, B_n)$

Rezultat operacije $r \div s$ je relacija na šemi

$$R - S = (A_1, \dots, A_m)$$

$$r \div s = \{ t \mid t \in \Pi_{R-S}(r) \wedge \forall u \in s (tu \in r) \}$$

Gde tu označava konkatenciju torki t i u u jednu torku



Deljenje (primer 1)

■ Relacije r, s :

A	B
α	1
α	2
α	3
β	1
γ	1
δ	1
δ	3
δ	4
ϵ	6
ϵ	1
β	2

B

1
2

s

■ $r \div s$:

A

α
β

r



Deljenje (primer 2)

■ Relacije r , s :

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

r

D	E
a	1
b	1

s

■ $r \div s$:

A	B	C
α	a	γ
γ	a	γ



Deljenje (Definicija Osn. Op. Rel. Alg.)

- Osobina
 - Neka je $q = r \div s$
 - Tada je q najveća relacija koja zadovoljava $q \times s \subseteq r$
- Definicija osnovnim operacijama relacione algebre
Neka su $r(R)$ i $s(S)$ relacije, i neka je $S \subseteq R$

$$r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r))$$

- $\Pi_{R-S,S}(r)$ preuređuje attribute relacije r
- $\Pi_{R-S}(\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r)$ daje one torke t u $\Pi_{R-S}(r)$ takve da za neke torke $u \in s$, $tu \notin r$.



Operacija Dodele

- Operacija dodele (\leftarrow) olakšava pisanje kompleksnih upita.
 - Pišemo upit kao sekvencijalni program koji se sastoji od
 - niza dodela
 - završno sa izrazom čija je vrednost rezultat upita.
 - Dodela se mora uvek vršiti u privremenu promenljivu relaciju.

- Primer: Napišimo $r \div s$ kao

$$temp1 \leftarrow \Pi_{R-S}(r)$$

$$temp2 \leftarrow \Pi_{R-S}((temp1 \times s) - \Pi_{R-S,S}(r))$$

$$result = temp1 - temp2$$

- Rezultat desno od \leftarrow je dodeljen promenljivoj relaciji na levoj strani \leftarrow .
- Mogu se koristiti promenljive u narednim izrazima.



Primeri (1)

- Find the names of all customers who have a loan and an account at bank.

$$\Pi_{customer_name}(borrower) \cap \Pi_{customer_name}(depositor)$$

- Find the name of all customers who have a loan at the bank and the loan amount

$$\Pi_{customer-name, loan-number, amount}(borrower \bowtie loan)$$



Primeri (2)

- Find all customers who have an account from at least the “Downtown” and the Uptown” branches.

- Query 1

$$\Pi_{customer_name} (\sigma_{branch_name = \text{“Downtown”}} (depositor \bowtie account)) \cap \Pi_{customer_name} (\sigma_{branch_name = \text{“Uptown”}} (depositor \bowtie account))$$

- Query 2

$$\Pi_{customer_name, branch_name} (depositor \bowtie account) \div \rho_{temp(branch_name)} (\{(\text{“Downtown”}), (\text{“Uptown”})\})$$

Note that Query 2 uses a constant relation.



Primeri (3)

- Find all customers who have an account at all branches located in Brooklyn city.

$$\Pi_{customer_name, branch_name} (depositor \bowtie account) \\ \div \Pi_{branch_name} (\sigma_{branch_city = \text{“Brooklyn”}} (branch))$$