

Relacijska algebra I

BAZE PODATAKA I

doc. dr. sc. Goran Oreški
*Fakultet informatike,
Sveučilište Jurja Dobrile, Pula*

Sadržaj

- ponavljanje prethodnih predavanja
 - terminologija baze podataka
 - relacijski model podataka
 - ključevi relacijskog modela
- jezici za rad s bazom podataka
 - ddl, dml
- relacijska algebra
 - osnovne operacije
 - primjeri
 - dodatne relacijske operacije
 - primjeri
- zadatak
- logički planovi upita

Ponavljjanje

- **terminologija baze podataka:**

- baza podataka – je organizirana kolekcija podataka (strukturiranih informacija) tipično spremljena elektronički na računalnom sustavu
- DBMS – software za upravljanje bazom podataka
- model podataka – je skup pravila koja određuju kako sve može izgledati logička struktura baze podataka
- vrste modela – mrežni, hijerarhijski, relacijski, objektni, polustrukturirani
- relacijski model podataka:
 - sastoji se od kolekcije tablica koje predstavljaju podatke i veze između podataka
 - tablice se nazivaju relacijama
 - svaka tablica definira n-torku određenog tipa koja se sastoji od fiksnog broja atributa

Ponavljjanje

- **relacijski model podataka:**

- relacija – naziv tablice u relacijskom modelu podataka
- atribut – jedan „stupac” relacije
- domena atributa - definira skup dozvoljenih vrijednosti za atribut
- n-torka - je uređeni skup parova naziv-vrijednost, fiksne veličine
- shema relacije - svaka relacija ima shemu, koja sadrži: uređeni skup atributa i domene za svaki atribut
- shema baze podataka - nacrt strukture baze podataka koja predstavlja logički dizajn baze podataka
- shema dijagram – grafički prikaz sheme baze podataka

Ponavljjanje

- **ključevi relacijskog modela:**

- ključevi - ključevi se koriste za razlikovanje pojedinih n-torki unutar relacije
- super ključ - je skup atributa koji zajedno, jednoznačno definiraju svaku n-torku unutar relacije
- kandidat ključ – minimalni super ključ
- primarni ključ - jedan od kandidata ključeva se odabire kao primarni način za jedinstveno identificiranje n-torki
- strani ključ – primarni ključ druge relacije
- ograničenja:
 - ograničenja primarnog ključa
 - ograničenja stranog ključa

Jezici za rad s bazom podataka

- dvije vrste jezika za rad s bazom podataka
 - DDL – *data-definition language*
 - specificiranje sheme baze podataka
 - DML – *data-manipulation language*
 - zadavanje upita i izmjenu podataka
- u praksi nisu odvojeni jezici već dva dijela jezika za rad s bazom podataka
- primjer SQL (*Structured Query Language*)

DML

- jezik koji omogućava korisnicima pristup i manipulaciju podataka organiziranih prema pripadajućem modelu podataka
- vrste pristupa su:
 - dohvat podataka pohranjenih u bazi podataka
 - umetanje novih podataka
 - brisanje podataka iz baze podataka
 - promjena već pohranjenih podataka
- dva tipa DML-a:
 - proceduralni
 - deklarativni

Tipovi DML-a

- proceduralni DML
 - zahtjeva od korisnika da odredi koje podatke i kako ih do njih doći
 - PL/SQL
- deklarativni DML
 - zvan i ne-proceduralni
 - zahtjeva od korisnika da odredi koje podatke treba bez specifikacije kako do njih doći
 - klasična SQL implementacija
- upitni (*engl. query*) jezik je dio DML zadužen za dohvat podataka iz baze podataka

DDL

- koristi se za specifikaciju sheme baze podataka
 - osim sheme mogu se specificirati i druga svojstva baze podataka
 - struktura pohrane i pristupne metode
- DDL omogućava postavljanje ograničenja na podacima:
 - domene – deklariranje atributa na neku domenu jest ograničenje na vrijednosti koje može poprimiti (*npr. integer, character, date*)
 - referencijalni integritet – provjera integriteta vrijednosti korištenih u stranim ključevima relacija
 - opća ograničenja (*engl. check constraint, assertions*) – uvjeti koje baza mora uvijek zadovoljavati (*npr. svaki smjer mora imati minimalno 20 kolegija*)
 - dozvole (*engl. authorization*) – prava korisnika za pristup podacima

Relacijska algebra

- proceduralni upitni jezik
- sastoji se od operacija relacijske algebre
 - šest osnovnih relacijskih operacija
- relacijske operacije:
 - dostavlja se jedna ili dvije relacije kao ulaz
 - stvara se relacija kao izlaz
- relacijske operacije se mogu kombinirati
 - svaka operacija stvara relaciju
 - upit je izraz relacijske algebre
 - druge korisne operacije se mogu stvoriti na temelju osnovnih

Zašto je korisna?

- SQL se ne temelji strogo na relacijskoj algebri !?
- SQL se više svrstava u deklarativni jezik !?
- relacijska algebra predstavlja matematičku notaciju
- mnogi sustavi baza podataka koriste operacije relacijske algebre za predstavljanje planova izvršenja (*je skup koraka kojih DBMS mora napraviti s ciljem da izvrši zadani upit*)
 - jednostavan i efektivan način za prikaz kako će rezultati biti generirani
 - relativno jednostavna manipulacija za optimizaciju upita

Osnovne operacije

- šest osnovnih operacija:

σ selekcija

Π projekcija

\cup unija

$-$ razlika

\times Kartezijev produkt

ρ preimenovanje

- svaka operacija
 - uzima jednu ili dvije relacije
 - stvara relaciju kao rezultat
- bitna pitanja
 - koje n-torke su uključene u relaciju rezultata?
 - postoje li ograničenja na ulaznim shemama?
 - koja je shema rezultata?

Operacija selekcije

- izraz: $\sigma_P(r)$
- P je predikat selekcije
 - P se može odnositi na atribut u r (ali ne iz druge relacije), i na konstantne vrijednosti (*engl. literals*)
 - mogu se koristiti operatori usporedbe: $=, \neq, <, \leq, >, \geq$
 - može se kombinirati više predikata $s : \wedge$ (and), \vee (or), \neg (not)
- r je relacija na ulazu
- relacija na izlazu sadrži sve n -torke za koje je P istina
- relacijska shema rezultata je ista kao i shema r

Primjer selekcije

- koristimo *nastavnik* relaciju

„Pronađite sve nastavnike koje predaju na odjelu informatike”

$\sigma_{\text{odjel}=\text{„Informatika”}}(\textit{nastavnik})$

„... s primanjima većim od 9000 kuna.”

$\sigma_{\text{odjel}=\text{„Informatika”} \wedge \text{primanja} > 9000}(\textit{nastavnik})$

id	prezime	odjel	primanja
10001	Knežević	Fizika	12000
10002	Horvat	Medicina	11000
10003	Kovačević	Informatika	8000
10004	Blažević	Književnost	4000
10011	Božić	Fizika	12000
20121	Novak	Kemija	10000
32100	Petrović	Informatika	9500

id	prezime	odjel	primanja
10003	Kovačević	Informatika	8000
32100	Petrović	Informatika	9500

id	prezime	odjel	primanja
32100	Petrović	Informatika	9500

Operacija projekcije

- izraz: $\Pi_{a,b,\dots}(r)$
- relacija na izlazu sadrži samo specificirane attribute iz r
 - specificirani atributi moraju se nalaziti u shemi r
 - shema relacije iz rezultata sadrži samo specificirane attribute
 - domene atributa su iste kao i domene atributa izvorne relacije r
- napomena:
 - relacija iz rezultata može imati manje redova nego izvorna relacija!
 - zašto?
 - relacije su skupovi n -torki, ne multiskupovi
 - oprez SQL!

Primjer projekcije

- koristimo *nastavnik* relaciju

id	prezime	odjel	primanja
10001	Knežević	Fizika	12000
10002	Horvat	Medicina	11000
10003	Kovačević	Informatika	8000
10004	Blažević	Književnost	4000
10011	Božić	Fizika	12000
20121	Novak	Kemija	10000
32100	Petrović	Informatika	9500

„Pronađite sve odjele na sveučilištu!”

$\Pi_{\text{odjel}}(\textit{nastavnik})$

- relacija iz rezultata ima 5 n-torki dok relacija ulaza ima 7
- shema relacije iz rezultata je (*odjel*)

odjel
Fizika
Medicina
Informatika
Književnost
Kemija

Složene operacije

- ulaz za operaciju može biti i izraz koji se vrednuje kao relacija a ne sama relacija
- $\Pi_{id}(\sigma_{\text{primanja} > 9000}(\textit{nastavnik}))$
 - dohvaća ID-eve svih nastavnika s primanjima većim od 9000 kuna
 - shema relacije na ulazu je:
 - $\textit{Nastavnik_shema} = (\underline{id}, \textit{prezime}, \textit{odjel}, \textit{primanja})$
 - shema relacije na izlazu:
 - (id)
 - razlika između osnovne i izvedene relacije
 - $\textit{nastavnik}$ je osnovna relacija (implementacija: tablica)
 - $\sigma_{\text{primanja} > 9000}(\textit{nastavnik})$ je primjer izvedene relacije (implementacija: upit, pogled)

Operacija unije

- izraz: $r \cup s$
- relacija na izlazu ima sve n -torke iz r i s
 - svaka n -torka je jedinstvena čak i ako se nalazi u r i s
- ograničenja na shemi za r i s ?
- r i s moraju imati kompatibilne sheme
 - r i s moraju imati isti stupanj relacije
 - za svaki atribut i u r i s , $r[i]$ mora imati istu domenu kao i $s[i]$
 - naziv atributa u relacijama nije bitan, relacije moraju zadovoljavati samo uvjete stupnja i domene

Primjer unije

- dodajmo relaciju *student* koja sadrži podatke o svim studentima na sveučilištu
- potrebno je pronaći sva prezimena koja se pojavljuju na sveučilištu (bez obzira da li se radi o studentima ili nastavnicima)
- možemo li koristiti operaciju unije za zadani zadatak?

id	prezime	odjel	primanja
10001	Kovačević	Fizika	12000
10002	Horvat	Medicina	11000
10003	Kovačević	Informatika	8000
10004	Blažević	Književnost	4000

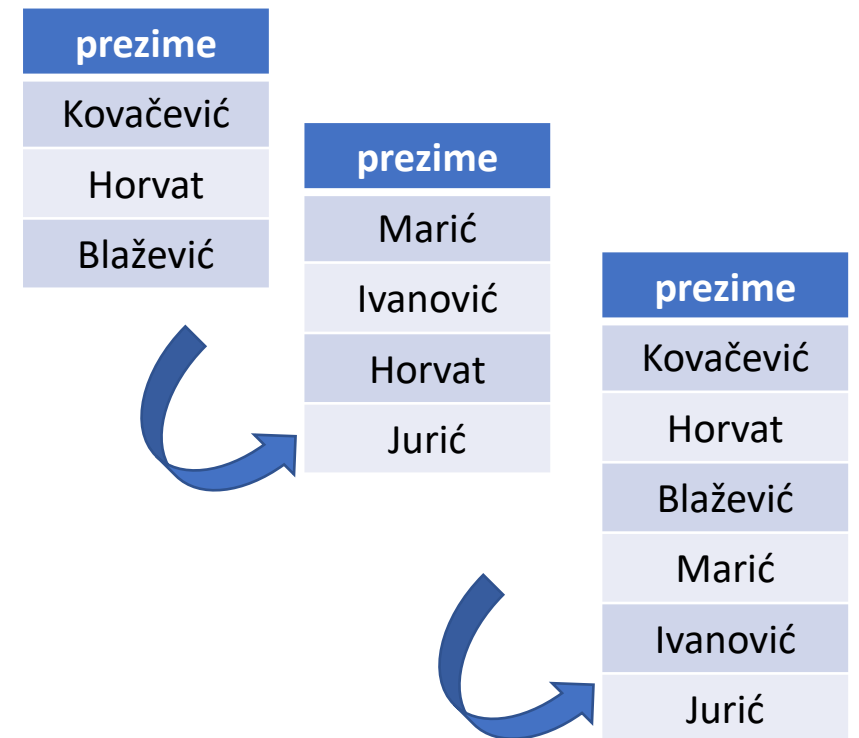
nastavnik

jmbag	prezime	smjer
100234	Marić	Biologija
203345	Ivanović	Geofizika
121455	Horvat	Informatika
200032	Jurić	Kardiologija

student

Primjer unije

- potrebno je pronaći sva prezimena koja se pojavljuju na sveučilištu (bez obzira da li se radi o studentima ili nastavnicima)
 - znamo pronaći prezimena profesora
 - $\Pi_{\text{prezime}}(\textit{nastavnik})$
 - prezimena studenata
 - $\Pi_{\text{prezime}}(\textit{student})$
 - rezultat je unija dva prethodna izraza
 - $\Pi_{\text{prezime}}(\textit{nastavnik}) \cup \Pi_{\text{prezime}}(\textit{student})$
 - uvjeti?
- iako ulazne relacije imaju 7 n-torki izlazna ima samo 6



Operacija razlike

- izraz: $r - s$
- rezultat sadrži n-torke koje se nalaze samo u r , ali ne i u s
 - n-torke koje su zajedničke r i s se isključuju
 - n-torke koje se nalaze samo u s se isključuju
- ograničenja na shemama r i s ?
 - sheme moraju biti kompatibilne
 - vrijede ista pravila kao i za operaciju unije

Primjer razlike

- potrebno je pronaći sve profesore koji ne dijele prezime s nijednim studentom na sveučilištu
- možemo li koristiti operaciju razlike za zadani zadatak?

id	prezime	odjel	primanja
10001	Kovačević	Fizika	12000
10002	Horvat	Medicina	11000
10003	Kovačević	Informatika	8000
10004	Blažević	Književnost	4000

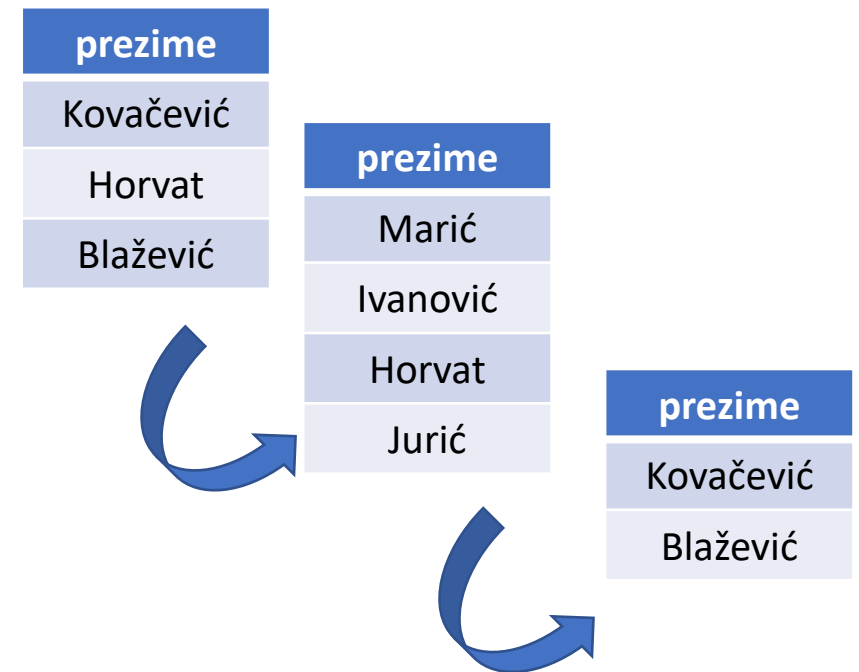
nastavnik

jmbag	prezime	smjer
100234	Marić	Biologija
203345	Ivanović	Geofizika
121455	Horvat	Informatika
200032	Jurić	Kardiologija

student

Primjer razlike

- potrebno je pronaći sve profesore koji ne dijele prezime s nijednim studentom na sveučilištu
 - znamo pronaći prezimena profesora
 - $\Pi_{\text{prezime}}(\textit{nastavnik})$
 - prezimena studenata
 - $\Pi_{\text{prezime}}(\textit{student})$
 - rezultat je unija dva prethodna izraza
 - $\Pi_{\text{prezime}}(\textit{nastavnik}) - \Pi_{\text{prezime}}(\textit{student})$
 - uvjeti?



Operacija Kartezijevog produkta

- izraz: $r \times s$ (*engl. r cross s*)
- nema ograničenja na sheme r i s
- shema rezultata jest spojena od shema relacija koje sudjeluju u izrazu
- slučaj kada shema r i s imaju attribute istog naziva
 - svi atributi istog naziva su uključeni, ne isključuje se niti jedan atribut
 - attribute istih naziva razlikujemo tako da im ispred imena dodamo naziv izvorišne relacije
- primjer
 - ulazne relacije: $r(a,b)$ i $s(b,c)$
 - shema $r \times s$, $(a, r.b, s.b, c)$

Operacija Kartezijevog produkta

- rezultat $r \times s$
 - sadrži sve n -torke iz r , kombinirano sa svim n -torkama iz s
 - ako r sadrži N n -torki, a s M n -torki, rezultat će sadržavati $N \times M$ n -torki
- operacija Kartezijevog produkta omogućava kombiniranje i uspoređivanje dvije relacije
 - želimo korelirati n -torke iz r relacije s n -torkama iz s relacije
 - način?
 - računamo $r \times s$, te potom odabiremo relacije koje zadovoljavaju određeni predikat

Kartezijev produkt primjer

- promijenimo shemu relacije *nastavnik*
- dodajmo relaciju *odjel*

id	prezime	primanja
10001	Knežević	12000
10002	Horvat	11000
10003	Kovačević	8000
10004	Blažević	4000

id	naziv	voditelj
211	Fizika	10002
212	Medicina	10004
213	Informatika	10003
214	Književnost	10001

- *nastavnik x odjel* - kardinalnost relacije?
 - $4 \times 4 = 16$

Kartezijev produkt primjer

- shema za *nastavnik*
 - *Nastavnik_shema = (id, prezime, primanja)*
- shema za *odjel*
 - *Odjel_shema = (id, naziv, voditelj)*
- shema za *nastavnik x odjel*
 - *(**nastavnik.id**, prezime, primanja, **odjel.id**, naziv, voditelj)*
 - atribut *id* se pojavljuje u obje relacije, zbog toga se u shemi navodi puno ime atributa koje sadrži i naziv relacije iz koje dolazi

Kartezijev produkt primjer

nastavnik.id	prezime	primanja	odjel.id	naziv	voditelj
10001	Knežević	12000	211	Fizika	10002
10001	Knežević	12000	212	Medicina	10004
10001	Knežević	12000	213	Informatika	10003
10001	Knežević	12000	214	Književnost	10001
10002	Horvat	11000	211	Fizika	10002
10002	Horvat	11000	212	Medicina	10004
10002	Horvat	11000	213	Informatika	10003
10002	Horvat	11000	214	Književnost	10001
10003	Kovačević	8000	211	Fizika	10002
10003	Kovačević	8000	212	Medicina	10004
10003	Kovačević	8000	213	Informatika	10003
10003	Kovačević	8000	214	Književnost	10001
10004	Blažević	4000	211	Fizika	10002
10004	Blažević	4000	212	Medicina	10004
10004	Blažević	4000	213	Informatika	10003
10004	Blažević	4000	214	Književnost	10001

Kartezijev produkt primjer

- može li se operacija Kartezijevog produkta koristiti za povezivanje redova iz dvije tablice?

nastavnik.id	prezime	primanja	odjel.id	naziv	voditelj
10001	Knežević	12000	211	Fizika	10002
10001	Knežević	12000	212	Medicina	10004
10001	Knežević	12000	213	Informatika	10003
10001	Knežević	12000	214	Književnost	10001
10002	Horvat	11000	211	Fizika	10002
10002	Horvat	11000	212	Medicina	10004
10002	Horvat	11000	213	Informatika	10003
10002	Horvat	11000	214	Književnost	10001

- kombiniranje Kartezijevog produkta sa selekcijom

$$\sigma_{\text{nastavnik.id} = \text{voditelj}} (\textit{nastavnik} \times \textit{odjel})$$

Kartezijev produkt primjer

- pronađite prezime voditelja odjela informatike na sveučilištu

id	prezime	primanja
10001	Knežević	12000
10002	Horvat	11000
10003	Kovačević	8000
10004	Blažević	4000

id	naziv	voditelj
211	Fizika	10002
212	Medicina	10004
213	Informatika	10003
214	Književnost	10001

- potrebne su relacije *nastavnik* i *odjel*
- potrebno je povezati n-torke u relaciji koristeći *id* nastavnika
- slijede jednostavne operacije relacijske algebre

Kartezijev produkt primjer

- povezivanje nastavnika i pripadajućih odjela pomoću Kartezijevog produkta i selekcije

$$\sigma_{\text{nastavnik.id} = \text{voditelj}} (\textit{nastavnik} \times \textit{odjel})$$

- odabir odjela informatike

$$\sigma_{\text{naziv} = \text{"informatika"}} (\sigma_{\text{nastavnik.id} = \text{voditelj}} (\textit{nastavnik} \times \textit{odjel}))$$

- ili

$$\sigma_{\text{nastavnik.id} = \text{voditelj} \wedge \text{naziv} = \text{"informatika"}} (\textit{nastavnik} \times \textit{odjel})$$

- projekcija relacije na prezime nastavnika

$$\Pi_{\text{prezime}} (\sigma_{\text{nastavnik.id} = \text{voditelj} \wedge \text{naziv} = \text{"informatika"}} (\textit{nastavnik} \times \textit{odjel}))$$

- rezultat?

Operacija preimenovanja

- rezultati relacijskih operatora nemaju ime
 - relacija koja nastaje kao rezultat ima shemu ali nema ime
- ime se može dodati koristeći operaciju preimenovanja
- izraz: $\rho_x(E)$ (*grčko slovo rho*)
 - E je izraz koji stvara relaciju
 - E može biti relacija s imenom ili varijabla relacije
 - x je novo ime relacije
- detaljniji izraz: $\rho_{x(A_1, A_2, \dots, A_n)}(E)$
 - imenuju se i svi atributi relacije
 - potrebno je da relacija bude stupnja n

Operacija preimenovanja

- operacije preimenovanja vrijedi isključivo u izrazu relacijske algebre u kojem se primjenjuje
 - operacija ne stvara novu relaciju
 - ime je vidljivo isključivo unutar izraza
- operator preimenovanja ima dvije osnovne funkcije
 - omogućava referenciranje izvedenih relacija i njihovih atributa unutar izraza relacijske algebre
 - omogućava osnovnoj relaciji unutar izraza da se koristi na više načina
 - $r \times \rho_s(r)$
- operator omogućava uklanjanje nejasnoća unutar izraza

Primjer preimenovanja

- pronađite *id* profesora koji ima najveću plaću

id	prezime	primanja
10001	Knežević	12000
10002	Horvat	11000
10003	Kovačević	8000
10004	Blažević	4000

nastavnik

- kako pronaći profesora s najvećom plaćom?
 - koristeći nama poznate operacije...
- znamo kako pronaći iznose manje od neke vrijednosti!
 - možemo li to iskoristiti?

Primjer preimenovanja

- kako pronaći sve profesore s plaćom manjom od nekog drugog profesora?
 - koristimo Kartezijev produkt relacije *nastavnik* sa sobom
nastavnik x nastavnik
 - usporedimo iznos plaće sa svim drugim iznosima
- problem?
 - ista imena relacija s lijeve i desne strane, problem s imenima atributa
 - rješenje?
 - korištenje operacije preimenovanja
nastavnik x ρ_{prof} (nastavnik)

Primjer preimenovanja

- potrebno je pronaći *ID* nastavnika koji imaju manje plaće od nekog drugog nastavnika

$$\Pi_{\text{nastavnik.id}}(\sigma_{\text{nastavnik.primanja} < \text{prof.primanja} (\text{nastavnik} \times \rho_{\text{prof}} (\text{nastavnik})))$$

- rezultat?

- potrebno je doći do najveće plaće

$$\Pi_{\text{id}} (\text{nastavnik}) - \Pi_{\text{nastavnik.id}}(\sigma_{\text{nastavnik.primanja} < \text{prof.primanja} (\text{nastavnik} \times \rho_{\text{prof}} (\text{nastavnik})))$$

- rezultat?

id
10001

- ukoliko postoji više profesora s najvišom plaćom?
 - svi id-evi će biti ispisani

Dodatne relacijske operacije

- osnovne operacije su dovoljne za izvršavanje upita na relacijskoj bazi podataka
- korištenje osnovnih operacija može proizvesti velike izraze za neke od osnovnih operacija (*npr. pronađi profesora s najvećom plaćom!*)
- postoje dodatne relacijske operacije koje se mogu definirati pomoću osnovnih

\cap presjek

\bowtie natural join (*hrv. prirodno spajanje*)

\div dijeljenje

\leftarrow dodjeljivanje

Operacija presjeka

- izraz: $r \cap s$
- $r \cap s = r - (r - s)$
 - $r - s$ = redovi koji su u r ali ne u s
 - $r - (r - s)$ = redovi koji se nalaze u r i s
- relacije moraju imati kompatibilne sheme
- primjer: pronađite sva zajednička prezimena nastavnika i studenata na sveučilištu

$$\Pi_{\text{prezime}}(\textit{nastavnik}) \cap \Pi_{\text{prezime}}(\textit{student})$$

Operacija natural join

- najčešća operacija Kartezijevog produkta je korelacija n-torki s istim vrijednostima ključa
 - join operacija (*hrv. operacija spajanja*)
- natural join je skraćena verzija te operacije
- izraz: $r \bowtie s$
 - r i s moraju imati zajednički/e attribute
 - najčešće se uzimaju ključevi od r ili s , ali nije nužno
 - zajednički atributi su oni koji imaju isti naziv u obje relacije

Operacija natural join

- uzmimo dvije relacije $r(R)$ i $s(S)$
- atributi koji se koriste za natural join
 - $R \cap S = \{A_1, A_2, A_3, \dots, A_n\}$
- formalna definicija:
 - $r \bowtie s = \Pi_{R \cup S} (\sigma_{r.A_1=s.A_1 \wedge r.A_2=s.A_2 \wedge r.A_3=s.A_3 \wedge \dots \wedge r.A_n=s.A_n} (r \times s))$
 - r i s su spojene koristeći uvjet jednakosti na zajedničkim atributima (prema nazivu)
 - rezultati su potom projicirani da bi se zajednički atributi pojavili samo jednom u konačnoj relaciji

Primjer natural join

- pronađite sve odjele zaposlenih nastavnika

id	prezime	odjel_naziv
10001	Knežević	Fizika
10002	Horvat	Književnost
10003	Kovačević	Informatika
10004	Blažević	Medicina
10011	Božić	Fizika
20121	Novak	Informatika
32100	Petrović	Informatika

nastavnik

odjel_naziv	voditelj
Fizika	Blažević
Medicina	Božić
Informatika	Novak
Književnost	Petrović

odjel

$\Pi_{\text{prezime}}(\sigma_{\text{nastavnik.odjel_naziv}=\text{odjel.odjel_naziv}}(\text{nastavnik} \times \text{odjel}))$



$\Pi_{\text{prezime}}(\text{nastavnik} \bowtie \text{odjel})$

Natural join karakteristike

- vrlo često se spaja, koristeći natural join, više od dvije tablice
- primjer:
 - $nastavnik \bowtie odjel \bowtie smjer$
- asocijativnost operacije natural join
 - $(nastavnik \bowtie odjel) \bowtie smjer$ je isto
 - $nastavnik \bowtie (odjel \bowtie smjer)$
- iako su izrazi isti, redoslijed operacija ima utjecaj na zahtjevnost izvršenja upita

Operacija dijeljenja

- izraz: $r \div s$
 - implementira „za svaki” tip upita
 - npr. „Pronađite sve redove u r koji imaju jedan red koji korespondira sa svim redovima u s .”
 - relacija r se dijeli s relacijom s
 - baza podataka ispita:
 - *polozeno(prezime, kolegij_naziv)*
 - pohranjuje se popis studenata koji su položili ispit
 - *kolegij(kolegij_naziv)*
 - pohranjuje se popis svih kolegija
- operacija koja najčešće nije zadano implementirana u SQL-u!*

Operacija dijeljenja

- pronajdi studente koji su položili sve ispite
 - cilj je pronaći studente u *polozeno* koji imaju unos za svaki kolegij iz tablice *kolegij*
 - potrebno je podijeliti *polozeno* s *kolegij*
- $$\textit{polozeno} \div \textit{kolegij}$$
- samo je student Horvat položio sve ispite
 - rezultat: *Horvat*

prezime	kolegij_naziv
Horvat	Programiranje 1
Horvat	Baze podatka 1
Božić	Sustavi poslovne inteligencije
Blažević	Programiranje 1
Božić	Baze podataka 1
Novak	Baze podataka 1
Horvat	Sustavi poslovne inteligencije

polozeno

kolegij_naziv
Programiranje 1
Baze podatka 1
Sustavi poslovne inteligencije

kolegij

Operacija dijeljenja

- za $r(R) \div s(S)$
 - potrebno je $R \subset S$
 - svi atributi iz S moraju se nalaziti i u R
 - rezultat je shema $R - S$
 - rezultat ima attribute koji su u R ali nisu u S
 - zbog tog razloga nije dozvoljeno $R = S$
 - svaka n -torka t iz rezultata zadovoljava slijedeće uvjete:
 - $t \in \Pi_{R-S}(r)$
 - $\langle \forall t_s \in s : \exists t_r \in r : t_r[S] = t_s[S] \wedge t_r[R - S] = t \rangle$
 - svaka n -torka u rezultatu ima red u r koji odgovara svakom redu u s

Operacija dijeljenja

- za $polozeno \div kolegij$
 - sheme su kompatibilne
 - rezultat ima shemu (*prezime*)
 - *prezime* se nalazi u *polozeno* ali ne u *kolegij*

prezime
Horvat

polozeno \div kolegij

- n-torka zadovoljava
 - $t \in \Pi_{R-S}(r)$
 - $\langle \forall t_s \in s : \exists t_r \in r : t_r[S] = t_s[S] \wedge t_r[R-S] = t \rangle$

prezime	kolegij_naziv
Horvat	Programiranje 1
Horvat	Baze podatka 1
Božić	Sustavi poslovne inteligencije
Blažević	Programiranje 1
Božić	Baze podataka 1
Novak	Baze podataka 1
Horvat	Sustavi poslovne inteligencije

polozeno

kolegij_naziv
Programiranje 1
Baze podatka 1
Sustavi poslovne inteligencije

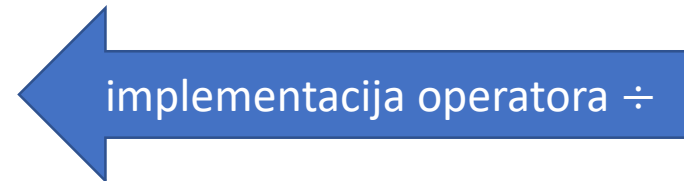
kolegij

Relacijske varijable

- relacijske varijable se odnose na specifičnu relaciju
 - to je specifičan skup n-torki s određenom shemom
- često se pod relacijom podrazumijeva relacijska varijabla (*relvar*)
 - mi smo do sada koristili naziv relacija, iako bi tehnički točnije koristiti relacijska varijabla
- slično kao u programskom jeziku:
int broj = 5;
 - broj je varijabla koja drži određenu vrijednost u nekom trenutku
 - relacijskoj varijabli se dodjeljuje relacijska vrijednost

Operacija dodjeljivanja

- operacija dodjeljivanja, pridružuje relacijsku vrijednost nekoj relacijskoj varijabli
- izraz: $relvar \leftarrow E$
 - E je izraz koji kao rezultat daje relaciju
- za razliku od ρ , $relvar$ ostaje u bazi podataka
- često se koristi za privremene relacijske varijable
 - $temp1 \leftarrow \Pi_{R-S}(r)$
 - $temp2 \leftarrow \Pi_{R-S}((temp1 \times s) - \Pi_{R-S,S}(r))$
 - $result \leftarrow temp1 - temp2$
 - izvođenje upita postaje niz koraka



Relacijska algebra - Zadatak

- zadane su relacije s pripadajućim shemama:

Dobavljac_shema = (db_id, db_naziv, adresa)

Dio_shema = (do_id, do_naziv, boja)

Katalog_shema = (db_id, do_id, cijena)

- opis zadanog modela:
 - *dobavljac* – pohranjuje podatke o tvrtkama koje dobavljaju rezervne dijelove
 - *dio* – pohranjuje podatke o rezervnim dijelovima
 - *katalog* – povezuje dvije relacije, koji dobavljač prodaje koji dio po kojoj cijeni

Relacijska algebra - Zadatak

- Zadatak 1.

- Pronađite nazive svih dobavljača koji isporučuju neki crveni dio!

$$\Pi_{db_naziv}((\sigma_{boja=„crvena”}(dio) \bowtie katalog) \bowtie dobavljac)$$

- Zadatak 2.

- Pronađite ID-eve svih dobavljača koji isporučuju neki crveni ili zeleni dio!

$$\Pi_{db_id}(\sigma_{boja=„crvena” \vee boja=„zelena”}(dio) \bowtie katalog)$$

- koristeći operaciju unije

$$\Pi_{db_id}(\sigma_{boja=„crvena”}(dio) \bowtie katalog \cup \sigma_{boja=„zelena”}(dio) \bowtie katalog)$$

- operacija projekcije unutar unije?

Relacijska algebra - Zadatak

- Zadatak 3.

- Pronađite ID-eve svih dobavljača koji isporučuju neki crveni dio ili im je adresa Rovinjska 14!

$$\Pi_{db_id}(\sigma_{boja=„crvena”(dio) \bowtie katalog) \cup \Pi_{db_id}(\sigma_{adresa=„Rovinjska 14”(dobavljac))$$

- Zadatak 4.

- Pronađite imena svih dobavljača koji isporučuju neki crveni dio ili im je adresa Rovinjska 14!

$$\Pi_{db_naziv}(\sigma_{boja=„crvena”(dio) \bowtie katalog \bowtie dobavljac) \cup \Pi_{db_naziv}(\sigma_{adresa=„Rovinjska 14”(dobavljac))$$

Relacijska algebra - Zadatak

- Zadatak 5.

- Pronađite ID-eve svih dobavljača koji isporučuju neki crveni i zeleni dio!

$$\Pi_{db_id}(\sigma_{boja=„crvena”(dio) \bowtie katalog) \cap \Pi_{db_id}(\sigma_{boja=„zelena”(dio) \bowtie katalog)$$

- Zadatak 6.

- Pronađite parove db id takve da dobavljač s prvim db id -om naplaćuje više za neki dio od dobavljača s drugim.

jefitniji \leftarrow katalog

skuplji \leftarrow katalog

$$\Pi_{db_id}(\sigma_{boja=„crvena”(dio) \bowtie katalog) \cap \Pi_{db_id}(\sigma_{boja=„zelena”(dio) \bowtie katalog)$$

Relacijska algebra - Zadatak

- Zadatak 7.

- Pronađite ID-eve svih dobavljača koji isporučuju samo crveni dio!

$$\Pi_{db_id}(dobavljac) - \Pi_{db_id}(katalog \bowtie \sigma_{boja \neq „crvena”}(dio))$$

- Zadatak 8.

- Pronađite ID-eve svih dobavljača koji isporučuju svaki dio!

$$\Pi_{db_id}(katalog \div \Pi_{do_id}(dio))$$

- *primjeri su samo jedna verzija rješenja, postoji mnogo drugih načina na koji mogu izraziti isti upiti pomoću relacijske algebre*
 - *domaća zadaća!*

Relacijska algebra - Zadatak

Što znače upiti relacijske algebre?

- Zadatak 1.

$$\Pi_{db_naziv}(\sigma_{boja=„crvena”}(dio) \bowtie \sigma_{cijena<100}(katalog) \bowtie dobavljac)$$

- pronađite naziv dobavljača koji isporučuju neki crveni dio za manje od 100 kn

- Zadatak 2.

$$\Pi_{db_naziv}(\Pi_{db_id}(\sigma_{boja=„crvena”}(dio) \bowtie \sigma_{cijena<100}(katalog)) \bowtie dobavljac)$$

- optimizirana verzija prošlog upita, Natural Join koristi projekciju *id dobavljača*

Relacijska algebra - Zadatak

- Zadatak 3.

$$\Pi_{db_naziv}(\sigma_{boja=„crvena”(dio) \bowtie \sigma_{cijena<100}(katalog) \bowtie dobavljac) \cap \Pi_{db_naziv}(\sigma_{boja=„crna”(dio) \bowtie \sigma_{cijena<100}(katalog) \bowtie dobavljac)$$

- pronađite dobavljače s istim imenom koji isporučuju crvene dijelove za manje od 100 kn i crne dijelove za manje od 100 kn
- ne mora biti isti dobavljač!
 - zašto?
 - kako osigurati da se radi o istom dobavljaču?

Relacijska algebra - Zadatak

- Zadatak 4.

$$\begin{aligned} \Pi_{db_id}(\sigma_{boja=„crvena”(dio) \bowtie \sigma_{cijena<100}(katalog) \bowtie dobavljac) \cap \Pi_{db_id}(\sigma_{boja=„crna”(dio) \\ \bowtie \sigma_{cijena<100}(katalog) \bowtie dobavljac) \end{aligned}$$

- pronađite *id-eve* dobavljača koji isporučuju crvene dijelove za manje od 100 kn i crne dijelove za manje od 100 kn
- recimo da je dodan strani ključ na atribut *katalog(db_id)*
 - razlika?
- da li se ovdje radi o istom dobavljaču?

Relacijska algebra - Zadatak

- Zadatak 5.

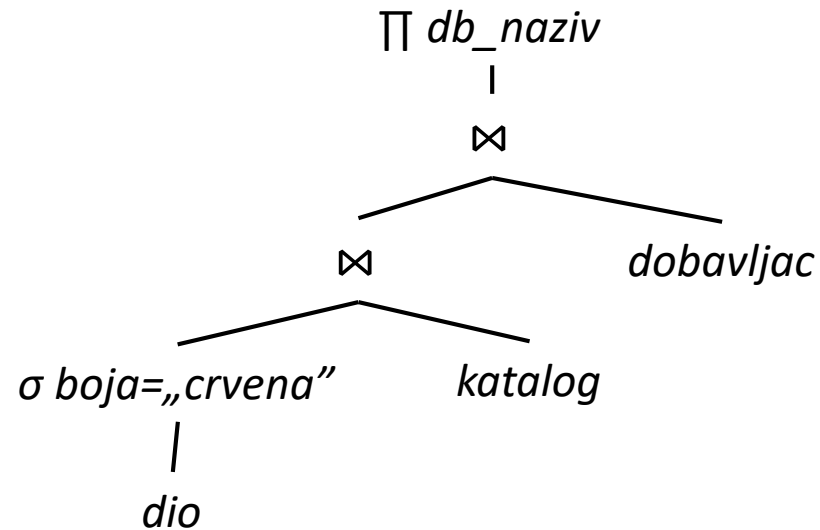
$$\Pi_{db_naziv}(\Pi_{db_id, db_naziv}((\sigma_{boja=„crvena”(dio) \bowtie \sigma_{cijena<100}(katalog) \bowtie dobavljac)) \cap \Pi_{db_id, db_naziv}((\sigma_{boja=„crna”(dio) \bowtie \sigma_{cijena<100}(katalog) \bowtie dobavljac))))$$

- pronađite imena dobavljača koji isporučuju crvene dijelove za manje od 100 kn i crne dijelove za manje od 100 kn
- da li se radi o istom dobavljaču?
 - zašto?

Logički planovi upita

- grafički prikaz izvođenja upita

- primjer: $\Pi_{db_naziv}((\sigma_{boja=„crvena”}(dio) \bowtie katalog) \bowtie dobavljac)$



Relacijska algebra

- varijante spomenutog join-a:
 - theta join – kombinacija selekcije i kartezijevog produkta
 - outer join operacije:
 - left outer join, right outer join, full outer join
- napredne operacije relacijske algebre:
 - generalizirana projekcija
 - agregacija
- povezane teme:
 - planovi izvođenja
 - optimizacija upita
 - ...



Literatura

- Pročitati
 - [DSC] poglavlje 1.4. – 1.5.
 - [DSC] poglavlje 6.1. (do 6.1.4.)
 - Caltech CS121
- Slijedeće predavanje
 - [DSC] poglavlje 6.1. (ostatak)
 - Caltech CS121