# Predavanje VIII.

*Optimizacija SQL upita*

BAZE PODATAKA II

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

# Sadržaj

- ponavljanje prethodnih predavanja
  - evaluacija upita
  - operacije relacijske algebre
  - izvođenje upita
- evaluacija složenih upita
- optimizacija upita
- pravila ekvivalencije
- primjer transformacije
- procjena troška
- odabir evaluacijskog plana

# Ponavljanje

- baze podataka obrađuju SQL upit kroz tri osnovna koraka:
  - parsiranje SQL u internu reprezentaciju plana
  - transformiranje interne reprezentacije u optimiziran plan izvršavanja
  - evaluacije optimiziranog plana
- planovi izvršavanja se u pravilu temelje na proširenoj relacijskoj algebri
- implementacija operacija:
  - selekcije
  - projekcije
  - sortiranje…

# Ponavljanje

- različite join implementacije
  - join ugniježđenom petljom
  - sort-merge join
  - hash join
- analiza izvršavanja upita
  - `EXPLAIN` naredba
- procjena troška izvršavanja za svaki upit
- statistika tablica
  - `ANALYZE TABLE` naredba

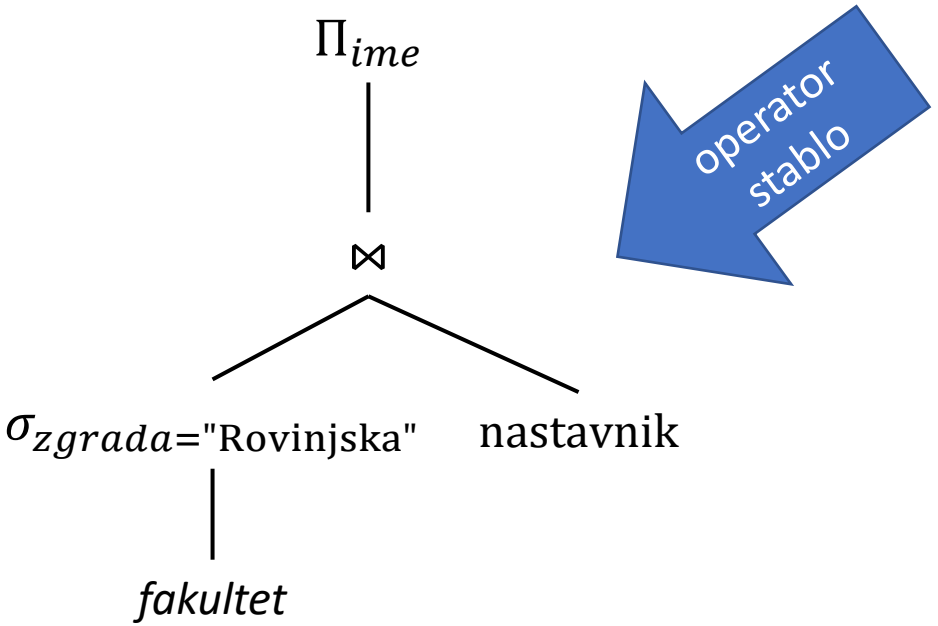
# Evaluacija upita

- do sada smo većinom proučavali kako se izvode individualne operacije relacijske algebre
- u posljednjem primjeru izvođenja upita, promatrali smo slučaj kada se kombinira više operacija (složeni *select*)
  - kako sustav provodi te operacije?
- jednostavan pristup
  - vrednovanje jedne po jedne operacije u ispravnom redoslijedu
  - rezultat svake operacije se materijalizira u privremenu tablicu za naknadno korištenje
  - nedostatak pristupa: svaki korak se zapisuje u privremenu tablicu, koja osim ako nije velika se zapisuje na disk

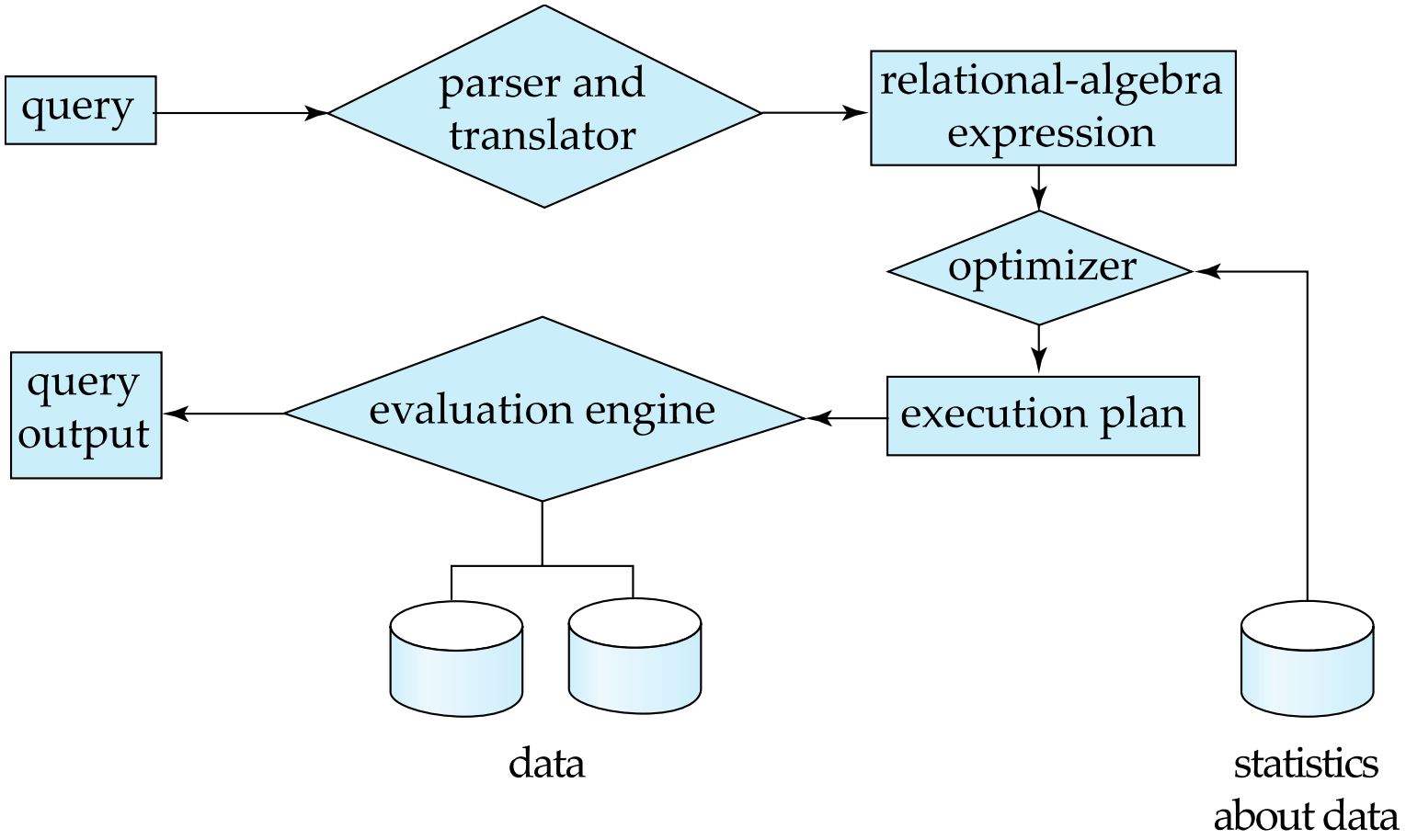
# Evaluacija upita

- alternativni pristup materijalizaciji je pipeline, gdje se rezultati jedne operacije odmah prosljeđuju drugoj bez potrebe da se pohranjuju
- promotrimo oba pristupa na primjeru:
  - materijalizirani
  - pipeline

$$\Pi_{ime}(\sigma_{\text{zgrada="Rovinjska"}}(fakultet) \bowtie nastavnik)$$



# Evaluacija upita



# Optimizacija upita

- na prošlom predavanju smo govorili o obradi upita i ukratko o optimizaciji upita
- optimizacija upita je proces odabira najučinkovitijeg plana za evaluaciju upita između mnogo strategija koje su moguće za određeni upit
  - broj alternativnih strategija raste sa složenosti upita
- očekujemo od sustava da stvori plan evaluacije upita koji minimizira trošak
- dva aspekta optimizacije:
  1. na razini relacijske algebre
  2. odabir detaljne strategije evaluacije

# Optimizacija upita

1. sustav će pokušati pronaći izraz koji je ekvivalentan traženom izrazu ali koji nosi manji trošak
2. detaljna strategija za obradu podrazumijeva: odabir algoritma za operaciju, odabir indeksa i sl.

- razlika u trošku (vremenu evaluacije) između dobre i loše strategije može biti značajna
- sustavu se isplati potrošiti dio vremena i analizirati strategije te odabrati najbolju
  - čak i ako se upit izvodi samo jednom

# Optimizacija upita

- generiranje planova evaluacije upita uključuje tri koraka:
  1. generiranje izraza koji su logički ekvivalentni zadanom izrazu
  2. označavanje generiranih izraza na različite načine da bi se dobili različiti planovi evaluacije
  3. odabir onog plana koji ima najmanji procijenjeni trošak
- procijenjeni trošak uključuje:
  - statistički podaci o relaciji
    - broj n-tokri, broj različitih vrijednosti po atributu
  - statistička procjena međurezultata
    - za računanje troška složenih izraza
  - trošak korištenih algoritama

# Optimizacija upita

*nastavnik(nastavnik_ID, ime, fakultet, primanja)*
*predaje(predaje_ID, kolegij_ID, nastavnik_ID, semestar, godina)*
*kolegij(kolegij_ID, fakultet, naziv, ECTS)*

- zadatak (napišite izraz relacijske algebre):
  - pronađite imena svih profesora na fakultetu Medicine zajedno s nazivom svih kolegija koje taj profesor predaje!

$$\Pi_{ime,naziv}(\sigma_{fakultet="Medicina"}(nastavnik \bowtie (predaje \bowtie \Pi_{kolegij\_id,naziv}(kolegij))))$$

    - ukoliko kolegij i nastavnik imaju atribut fakultet?
      - projekcija zbog natural join-a
  - velika relacija (međurezultat) kao rezultat join-a

$$nastavnik \bowtie predaje \bowtie \Pi_{kolegij\_id,\, naziv}(kolegij)$$

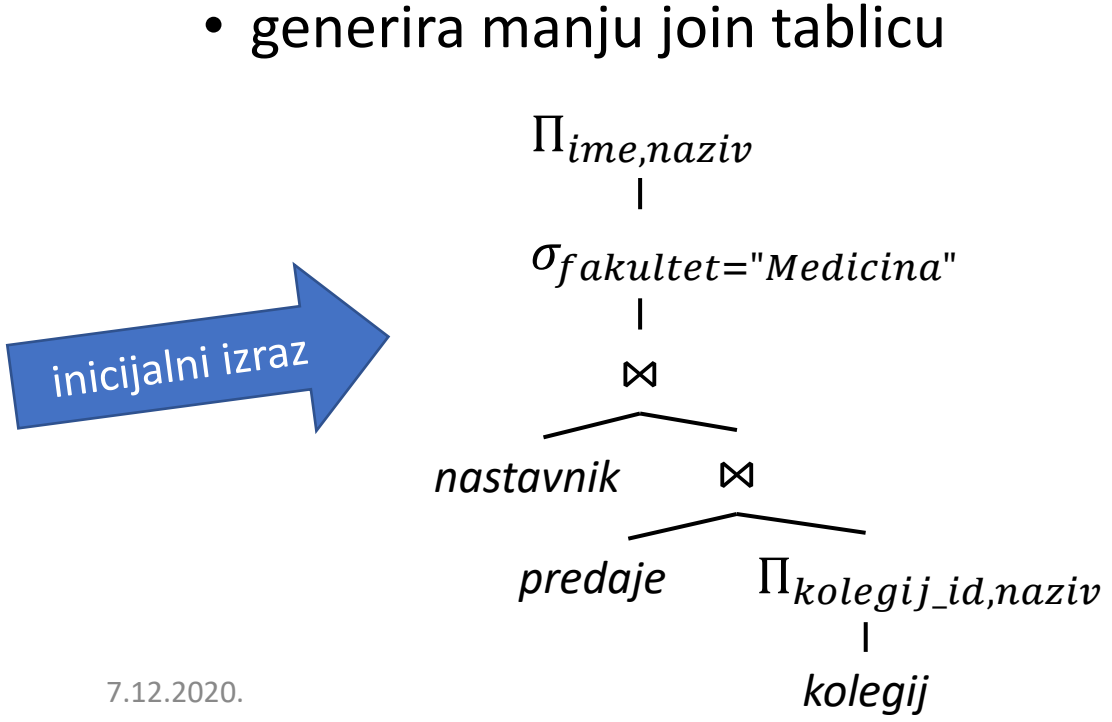
    - zanimaju nas samo n-torke koje se odnose na profesore Medicine
    - i to svega dva atributa
    - možemo li sastaviti drugačiji izraz uzevši to u obzir?

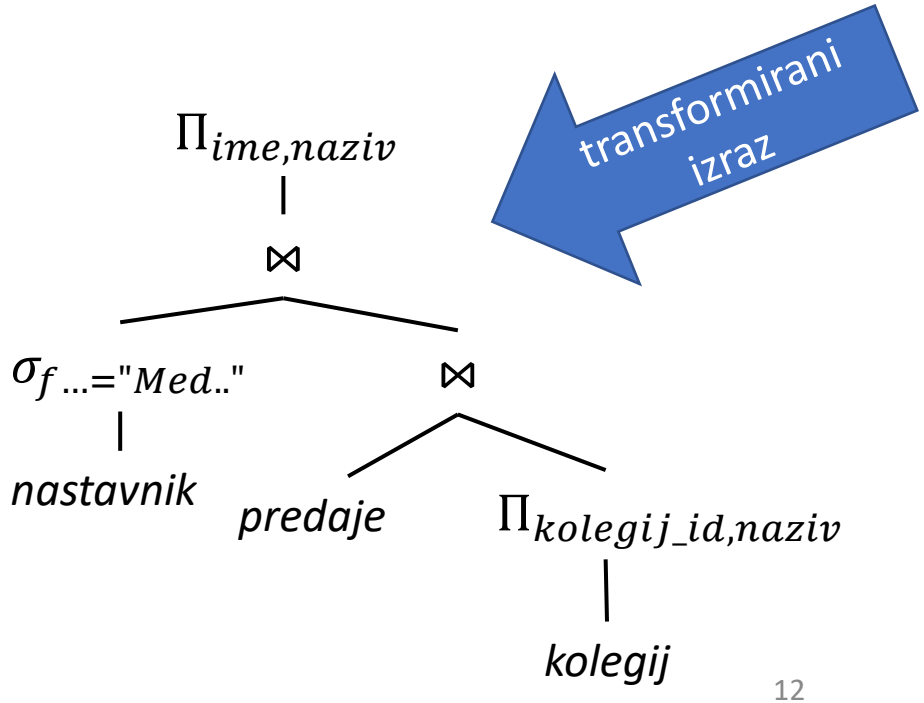
# Optimizacija upita

- novi izraz:

$$\Pi_{ime,naziv}((\sigma_{fakultet="Medicina"}(nastavnik)) \bowtie (predaje \bowtie \Pi_{kolegij\_id,naziv}(kolegij)))$$

  - ekvivalentan prvom izrazu
  - generira manju join tablicu



# Pravila ekvivalencije

- logički ekvivalentni izrazi se temelje na pravilima ekvivalencije
- tj. pravila ekvivalencije definiraju ekvivalentnost dva izraza
  - generiraju isti rezultat
    - redoslijed n-torki nije bitan
  - ekvivalentna ali ne i jednako skupa!
- ekvivalentan pravila u:
  - relacijskoj algebri -> generiraju isti skup
  - SQL -> generiraju isti multiskup
- optimizator koristi pravila ekvivalencije za transformaciju izraza

# Pravila ekvivalencije

- primjeri pravila:
  1. konjuktivna operacije selekcije može biti rastavljena u niz pojedinačnih selekcija

$$\sigma_{\theta_1 \wedge \theta_2}(E) = \sigma_{\theta_1}(\sigma_{\theta_2}(E))$$

  2. operacija selekcije je komutativna

$$\sigma_{\theta_2}(\sigma_{\theta_1}(E)) = \sigma_{\theta_1}(\sigma_{\theta_2}(E))$$

  3. smo je posljednja operacija projekcije u nizu dovoljna, prethodne nisu potrebne

$$\Pi_{L_1}(\Pi_{L_2}(\dots(\Pi_{L_n}(E)))) = \Pi_{L_1}(E)$$

# Pravila ekvivalencije

- primjeri pravila:
  4. selekcije se mogu kombinirati s Kartezijevim produktom i theta join-om

$$\sigma_{\theta}(E_1 \times E_2) = E_1 \bowtie_{\theta} E_2$$
$$\sigma_{\theta_1}(E_1 \bowtie_{\theta_2} E_2) = E_1 \bowtie_{\theta_1 \wedge \theta_2} E_2$$

  5. theta i natural join operacije su komutativne

$$E_1 \bowtie_{\theta} E_2 = E_2 \bowtie_{\theta} E_1$$

  6. pravilo
     a) natural join operacije su asocijativne

$$(E_1 \bowtie E_2) \bowtie E_3 = E_1 \bowtie (E_2 \bowtie E_3)$$

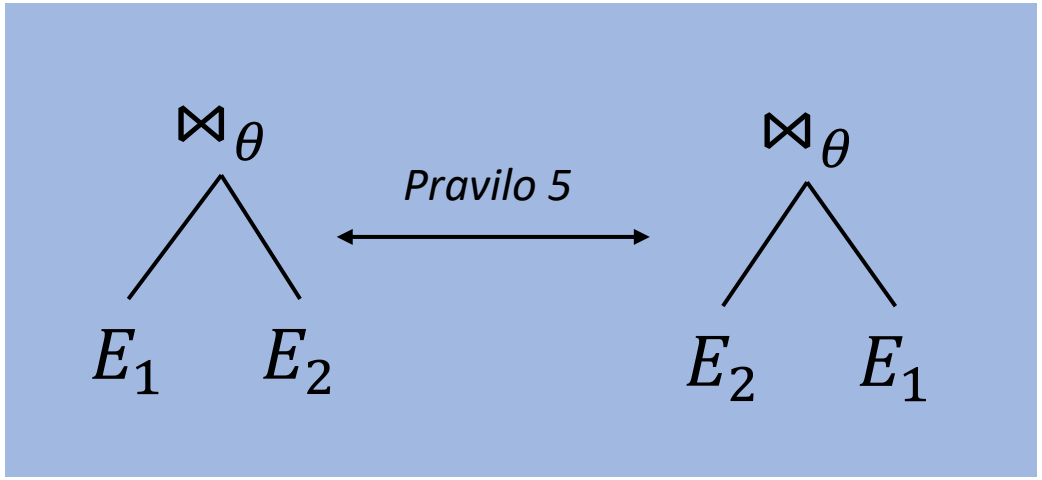
     b) theta join operacije pod slijedećim uvjetom

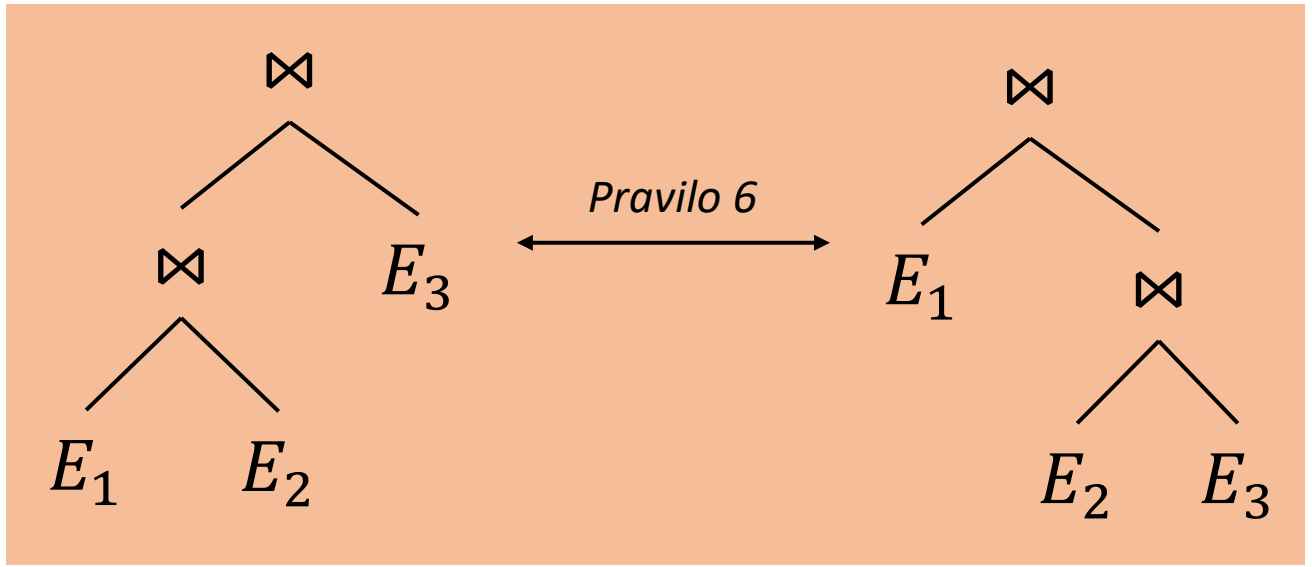
$$(E_1 \bowtie_{\theta_1} E_2) \bowtie_{\theta_2 \wedge \theta_3} E_3 = E_1 \bowtie_{\theta_1 \wedge \theta_3} (E_2 \bowtie_{\theta_2} E_3)$$

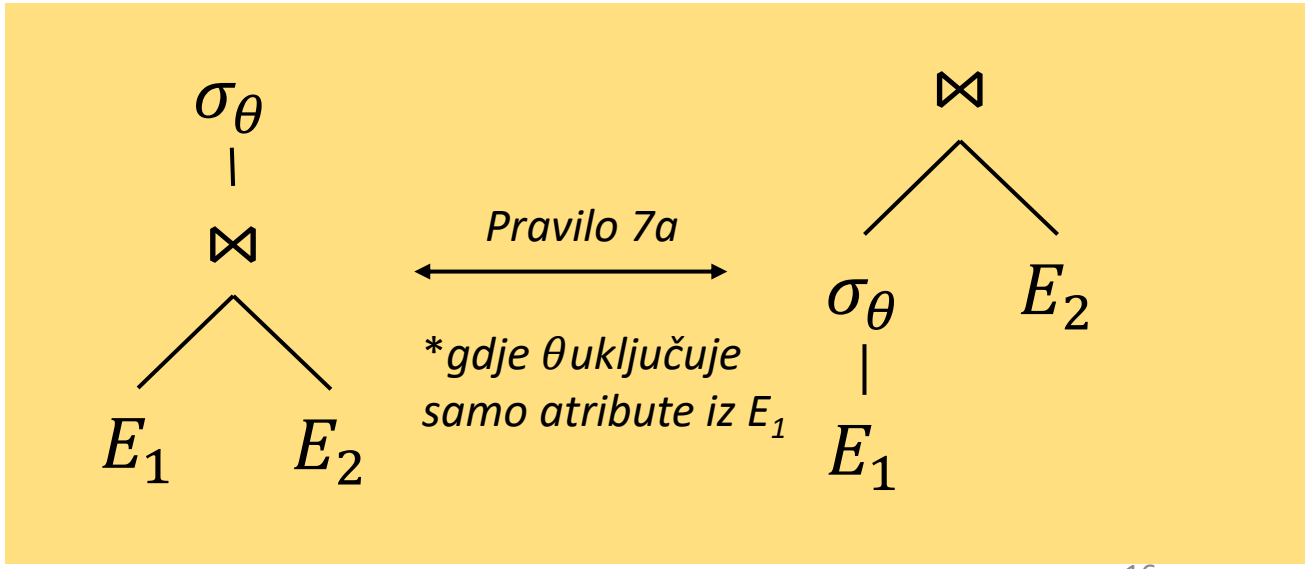
**gdje $\theta_2$ uključuje atribute samo iz $E_2$ i $E_3$*

# Pravila ekvivalencije

- grafički prikaz pravila




***DZ Nacrtati sva pravila u Powerpointu – prva dva ispravna rješenja 2 boda***

# Pravila ekvivalencije

- primjeri pravila:
  7. operacija selekcije se distribuira preko theta join operacije pod uvjetima
     a) svi atributi unutar $\theta_0$ se odnose samo na atribute iz jednog izraza koji se spaja

$$\sigma_{\theta_0}(E_1 \bowtie_\theta E_2) = (\sigma_{\theta_0}(E_1)) \bowtie_\theta E_2$$

     b) svi atributi unutar $\theta_1$ se odnose samo na atribute iz $E_1$ a svi atributi unutar $\theta_2$ se odnose samo na atribute iz $E_2$

$$\sigma_{\theta_1 \wedge \theta_2}(E_1 \bowtie_\theta E_2) = (\sigma_{\theta_1}(E_1)) \bowtie_\theta (\sigma_{\theta_2}(E_2))$$

  8. operacija projekcije se distribuira preko theta join operacije pod uvjetima
     a) ako $\theta$ uključuje samo atribute iz $L_1$ unija $L_2$

$$\Pi_{L_1 \cup L_2}(E_1 \bowtie_\theta E_2) = (\Pi_{L_1}(E_1)) \bowtie_\theta (\Pi_{L_2}(E_2))$$

# Pravila ekvivalencije

8. operacija projekcije se distribuira preko theta join operacije pod uvjetima

   b) uzmimo u obzir join $E_1 \bowtie_\theta E_2$

   - neka su $L_1$ i $L_2$ skup atributa iz $E_1$ i $E_2$ respektivno
   - neka su $L_3$ atributi iz $E_1$ koji sudjeluju u join uvjetu $\theta$ ali ne sudjeluju u $L_1$ unija $L_2$ i
   - neka su $L_4$ atributi iz $E_2$ koji sudjeluju u join uvjetu $\theta$ ali ne sudjeluju u $L_1$ unija $L_2$ i

$$\Pi_{L_1 \cup L_2}(E_1 \bowtie_\theta E_2) = \Pi_{L_1 \cup L_2}((\Pi_{L_1 \cup L_3}(E_1)) \bowtie_\theta (\Pi_{L_2 \cup L_4}(E_2)))$$

9. operacije unije i presjeka su komutativne

$$E_1 \cup E_2 = E_2 \cup E_1$$
$$E_1 \cap E_2 = E_2 \cap E_1$$

   - operacija razlike nije!

# Pravila ekvivalencije

10. operacije unije i presjeka su asocijativne

$$(E_1 \cup E_2) \cup E_3 = E_1 \cup (E_2 \cup E_3)$$
$$(E_1 \cap E_2) \cap E_3 = E_1 \cap (E_2 \cap E_3)$$

11. operacije selekcije se distribuira preko operacije unije, presjeka i razlike

$$\sigma_\theta(E_1 - E_2) = \sigma_\theta(E_1) - \sigma_\theta(E_2)$$

- vrijedi za unije i presjek

$$\sigma_\theta(E_1 - E_2) = \sigma_\theta(E_1) - E_2$$

  - vrijedi za presjek ali ne i uniju

12. operacije projekcije se distribuira preko operacije unije

$$\Pi_L(E_1 \cup E_2) = (\Pi_L(E_1)) \cup (\Pi_L(E_2))$$

- *postoje i druga pravila ekvivalencije koja obuhvaćaju operacije proširene relacijske algebre*

# Primjeri transformacije

- korištenje pravila ekvivalencije
- na shemi sveučilišta:

  *nastavnik(nastavnik_ID, ime, fakultet, primanja)*
  *predaje(predaje_ID, kolegij_ID, nastavnik_ID, semestar, godina)*
  *kolegij(kolegij_ID, fakultet, naziv, ECTS)*

- primjer s početka predavanja

$$\Pi_{ime,naziv}(\sigma_{fakultet="Medicina"}(nastavnik \bowtie (predaje \bowtie \Pi_{kolegij\_id,naziv}(kolegij))))$$

- transformiran

$$\Pi_{ime,naziv}((\sigma_{fakultet="Medicina"}(nastavnik)) \bowtie (predaje \bowtie \Pi_{kolegij\_id,naziv}(kolegij)))$$

  - koje pravilo je korišteno u transformaciji?
    - 7a

# Primjeri transformacije

- više pravila ekvivalencije se mogu koristiti jedno iza drugog na cijelom ili djelu upita
- zadatak (napišite izraz relacijske algebre):
  - pronađite imena svih profesora na fakultetu Medicine koji su predavali 2018. godine zajedno s nazivom svih kolegija koje taj profesor predaje!

$$\Pi_{ime,naziv}(\sigma_{fakultet="Medicina" \wedge godina=2018}(nastavnik \bowtie (predaje \bowtie \Pi_{kolegij\_id,naziv}(kolegij))))$$

  - ne možemo primijeniti predikat selekcije direktno na relaciju nastavnik jer uključuje atribute iz obje relacije, nastavnik i predaje
  - možemo primijeniti pravilo asocijativnosti 6a

$$\Pi_{ime,naziv}(\sigma_{fakultet="Medicina" \wedge godina=2018}((nastavnik \bowtie predaje) \bowtie \Pi_{kolegij_{id},naziv}(kolegij)))$$

# Primjeri transformacije

- korištenjem pravila 7.a možemo upit dalje raspisati

$$\Pi_{ime,naziv}((\sigma_{fakultet="Medicina" \wedge godina=2018}(nastavnik \bowtie predaje)) \bowtie \Pi_{kolegij_{id},naziv}(kolegij))$$

- primjenom pravila 1 možemo razbiti podizraz na slijedeći način

$$\sigma_{fakultet="Medicina"}\,(\sigma_{godina=2018}(nastavnik \bowtie predaje)$$

- nova prilika za pravilo 7.a (primjeni selekcije rano)

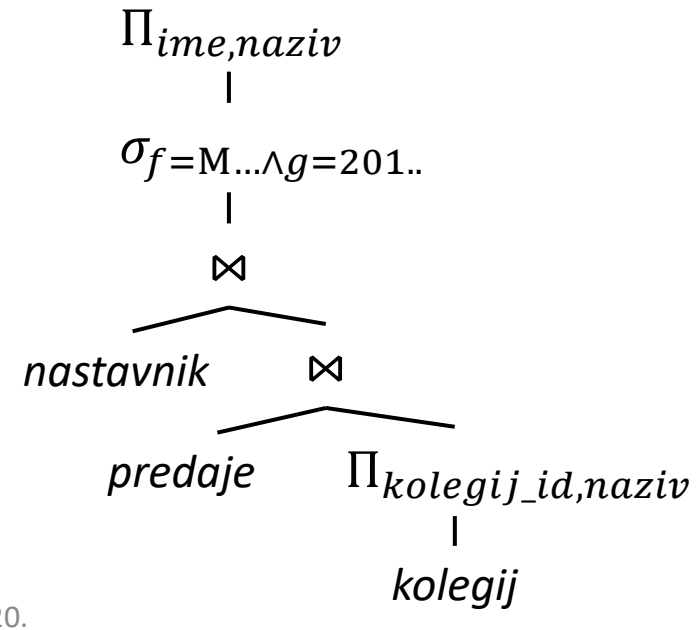
$$\sigma_{fakultet="Medicina"}\,(nastavnik) \bowtie \sigma_{godina=2018}(predaje)$$

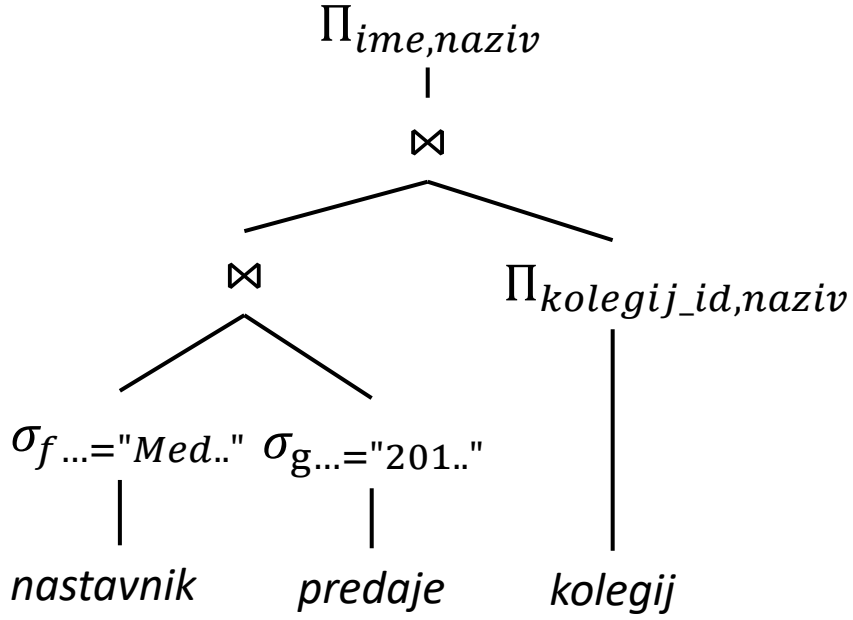
# Primjer transformacije

- grafički prikaz inicijalnog i transformiranog izraza

$$\Pi_{ime,naziv}(\sigma_{fakultet="Medicina" \wedge godina=2018}(nastavnik \bowtie (predaje \bowtie \Pi_{kolegij\_id,naziv}(kolegij))))$$

$$\Pi_{ime,naziv}(\sigma_{fakultet="Medicina"}(nastavnik) \bowtie \sigma_{godina=2018}(predaje) \bowtie \Pi_{kolegij_{id},naziv}(kolegij))$$



# Primjer transformacije

- pogledajmo izraz

$$\Pi_{ime,naziv}((\sigma_{fakultet="Medicina"}(nastavnik) \bowtie predaje) \bowtie \Pi_{kolegij\_id,naziv}(kolegij))$$

- kada računamo podizraz

$$(\sigma_{fakultet="Medicina"}(nastavnik) \bowtie predaje)$$

- kao rezultat dobivamo shemu

(*nastavnik_ID, ime, fakultet, primanja, predaje_ID, kolegij_ID, semestar, godina*)

- nisu svi atributi potrebni

$$\Pi_{ime,naziv}((\Pi_{kolegij\_id,\text{ime}}(\sigma_{fakultet="Medicina"}(nastavnik) \bowtie predaje)) \bowtie \Pi_{kolegij\_id,naziv}(kolegij))$$

# Primjer transformacije

- join raspored je bitan za smanjivanje privremenih rezultata
  - optimizatori jako paze na join raspored
  - pravilo 6.a (asocijativne join operacije)

$$(E_1 \bowtie E_2) \bowtie E_3 = E_1 \bowtie (E_2 \bowtie E_3)$$

  - iako su izrazi ekvivalentni, trošak može biti značajno različit
- na primjer:

$$\Pi_{ime,naziv}((\sigma_{fakultet="Medicina"}(nastavnik)) \bowtie predaje \bowtie \Pi_{kolegij\_id,naziv}(kolegij))$$

  - ispravni redoslijed?
    - predaje i proj. kolegij?
    - selek. nastavnik i predaje?

# Primjer transformacije

- natural join je komutativan (pravilo 5)

$$E_1 \bowtie E_2 = E_2 \bowtie E_1$$

- primjer:

$$(nastavnik \bowtie \Pi_{kolegij\_id,naziv}(kolegij)) \bowtie predaje$$

- postoji li bolje rješenje?
  - nastavnik i kolegij nemaju zajedničkih kolona
  - rezultat Kartezijev produkt
    - loš odabir

$$(nastavnik \bowtie predaje) \bowtie \Pi_{kolegij\_id,naziv}(kolegij)$$

# Pronalaženje ekvivalentnih izraza

- optimizatori upita koriste pravila ekvivalencije da bi generirali izraze koji su ekvivalentni zadanom
- pravila se mogu generirati na slijedeći način:
  - ako je zadan izraz *E*, skup ekvivalentnih izraza *EQ* na početku sadrži samo *E*
  - svaki izraz unutar *EQ* se uparuje sa svim pravilima ekivalencije na način:
    - ukoliko bilo koji podizraz unutar promatranog izraza odgovara jednoj strani pravila ekvivalencije, stvara se novi izraz su koja uključuje drugu stranu pravila
    - novi izraz se dodaje u *EQ*
  - postupak se nastavlja sve kod se mogu generirati novi izrazi
- ovaj pristup je vrlo skup po pitanju prostora i vremena

# Pronalaženje ekvivalentnih izraza

- optimizacija transformacije izraza
- troškovi se mogu smanjiti koristeći slijedeće ideje:
  - mnogi izrazi dijele zajedničke podizraze, stoga se koristi tehnika reprezentacije koja dopušta više izraza da pokazuju na dijeljeni podizraz i na taj način smanje prostorni trošak
  - ne moraju se uvjek istraživati svi podizrazi, već je moguće uzeti u obzir trošak nekih izraza, optimizator u tu svrhu može koristiti određene tehnike koje će smanjiti vremenski trošak
    - dinamičko programiranje
    - heuristike

# Procjena troška

- trošak operacije ovisi o veličini i ostalim statistikama podataka na ulazu
  - npr. trošak izraza $a \bowtie (b \bowtie c)$ i spajanja *a* s rezultatom spoja *b* i *c* ovisi o veličini i drugim statistikama tog spoja
- statistike za svaku relaciju su pohranjene u katalozima sustava baze podataka
- procjena troška je samo procjena!
  - ne mora biti precizna jer se može temeljiti na krivim pretpostavkama
  - stvarni trošak != procijenjeni trošak
  - najmanji procijenjeni trošak ne mora biti i najmanji stvarni, iako u pravilu je

# Procjena troška

- informacije kataloga:
  - broj n-torki u relaciji
  - broj blokova relacije
  - veličina n-torke u bajtovima
  - broj n-torki relacije koje stanu u jedan blok
  - broj različitih vrijednosti za pojedini stupac relacije
    - ili skup stupaca
- katalozi sadrže informacije i o indeksima
  - dubina btree stabla
  - broj listova

# Procjena troška

- navedeni statistički podaci predstavljaju pojednostavljenje u odnosu na informacije koje održavaju stvarni sustavi
- npr. podaci o distribuciji vrijednosti atributa se održavaju pomoću histograma, koji dijeli sve vrijednosti u određeni broj razreda
- histogram također često sadrži i broj jedinstvenih vrijednosti po razredima
- konceptualno statistika tablica se može promatrati kao materijalizirani pogledi, održavanje može biti jako skupo
  - uzorci podataka
  - održavanje na temelju odluke DB administratora

# Procjena troška

- procjena troška uključuje procjenu veličine rezultata
- procjena veličine se donosi na temelju korištenih operacija:
  - trošak selekcije ovisi  o korištenom predikatu
    - jednakost, raspon, složen
  - trošak join-a
    - jednostavno računanje Kartezijevog produkta
    - druge join operacije ovise o presjeku shema relacija, tj. atributima koji su uključeni u presjek
  - ostale operacije
    - projekcija
    - agregacija
    - operacije skupova

# Odabir evaluacijskog plana

- do sada smo vidjeli kako se generiraju ekvivalentni izrazi te što se koristi prilikom računanja procjene troška nekog upita
- svaka operacija unutar generiranih izraza se može implementirati pomoću drugog algoritma
- **evaluacijski plan** definira točno koji algoritam će se koristiti za koju operaciju te kako će izvršavanje operacija biti koordinirano
- jednom kada imamo evaluacijski plan možemo procijeniti trošak izvođenja na temelju
  - statistike relacije
  - troškova pojedinih algoritama

# Odabir evaluacijskog plana

- prilikom odabira plana potrebno je uzeti u obzir interakciju evaluacijskih tehnika
  - odabir najboljeg algoritma za pojedinu operaciju ne mora značiti i najefikasniji plan izvođenja
    - merge-join može biti skuplji od hash-joina ali rezultira s sortiranim rezultatima koji mogu biti korisni kasnijim operacijama
    - ugniježđeni join može pružiti priliku korištenja pipeline-a
- optimizatori uključuju elemente dva široka pristupa:
  - pretraži sve planove i odaberi najbolji na temelju troška (cost-based optimizer)
  - koristi heuristiku za odabir plana

# Odabir evaluacijskog plana

- primjer, pronalazak najboljeg poretka join naredbe

$$r_1 \bowtie r_2 \bowtie \cdots \bowtie r_n$$

- postoji *(2(n − 1))!/(n − 1)!* različitih join poredaka, n=3?

$r_1 \bowtie (r_2 \bowtie r_3)$ $r_1 \bowtie (r_3 \bowtie r_2)$ $(r_2 \bowtie r_3) \bowtie r_1$ $(r_3 \bowtie r_2) \bowtie r_1$
$r_2 \bowtie (r_1 \bowtie r_3)$ $r_2 \bowtie (r_3 \bowtie r_1)$ $(r_1 \bowtie r_3) \bowtie r_2$ $(r_3 \bowtie r_1) \bowtie r_2$
$r_3 \bowtie (r_1 \bowtie r_2)$ $r_3 \bowtie (r_2 \bowtie r_1)$ $(r_1 \bowtie r_2) \bowtie r_3$ $(r_2 \bowtie r_1) \bowtie r_3$

  - za n = 3 -> 12
  - za n = 7 -> 665280
  - za n = 10 -> 17643225600

# Odabir evaluacijskog plana

- nema potrebe za generiranjem svih kombinacija
- možemo koristiti dinamičko programiranje
  - pohranjivanje rezultata računanja da bi se naknadno koristili
  - može značajno smanjiti vrijeme računanja
- primjer:

$$(r_1 \bowtie r_2 \bowtie r_3) \bowtie r_4 \bowtie r_5$$

  - kombinacije bez pohranjivanja rezultata
    - n=6 -> 144
  - kombinacije s pohranjivanjem rezultata
    - n=6 -> 24

# Odabir evaluacijskog plana

- ponekad je troškovna optimizacija čak i s dinamičkim programiranjem preskupa
- sustav može koristiti heuristike da bi smanjivo vrijeme pretrage prostora rješenja
- heuristička transformacija se temelji na skupu pravila koji u pravilu poboljšavaju performanse (ali ne uvjek):
  - izvodi selekcije rano (smanjuje se broj n-torki)
  - izvodi projekcije rano (smanjuje broj atributa)
  - izvodi najrestriktivnije selekcije i join operacije rano prije drugih sličnih operacija
- neki sustavi koriste samo heuristike dok ih drugi kombiniraju s t. o.

# Literatura

- Pročitati
  - [DSC] poglavlje 13.1. – 13.5.
- Slijedeće predavanje
  - [DSC] poglavlje 14.