



# Zadatci za vježbu

## #12

Previous step

Submit answer

Next step

Layout ▼

Tutorial content ▼

### 3/4: Heuristička optimizacija

Budući da su u definiciji shema relacija ispuštene naredbe za definiranje primarnih i stranih ključeva, SUBP nema informaciju o ključevima relacija, niti su stvoreni indeksi koji se u Postgresu automatski kreiraju prilikom kreiranja primarnog ključa.

Optimizatoru su na raspolaganju sljedeći podaci:	
N (mjestoOp)	1000
N (studentOp)	30
N (ispitOp)	1000
V (JMBAG, studentOP)	30
V (JMBAG, ispitOP)	15
V (pbr, mjestoOP)	1000
V (datumIspit, ispitOp)	600
V (pBrRodStudent, studentOp)	15
V (pBrStanStudent, studentOp)	15
V (sifPred, ispitOp)	15

Za SELECT naredbe u primjerima 1. - 4. potrebno je nacrtati plan izvršavanja nakon provedene heurističke optimizacije.

Redoslijed spajanja odrediti na temelju procjene ukupnog broja n-torki u svim međurezultatima.

U planu naznačiti očekivani broj zapisa za međurezultate te korištene metode pristupa.

Navesti sve izraze prema kojima je obavljena procjena broja n-torki u međurezultatima.

#### Primjer 1.

```
SELECT *
FROM ispitOp, studentOp, mjestoOp
WHERE ispitOp.JMBAG = studentOp.JMBAG
      AND studentOp.pBrRodStudent = mjestoOp.pbr
      AND ispitOp.datumIspit = '11.7.2020';
```

rješenje

#### Određivanje veličine međurezultata:

$ispitOP2 = \sigma_{datumIspit='11.07.2020'}(ispitOp)$

$N(ispitOP2) = N(ispitOp) / V(datumIspit, ispitOp) = 1000 / 600 = 1.7$

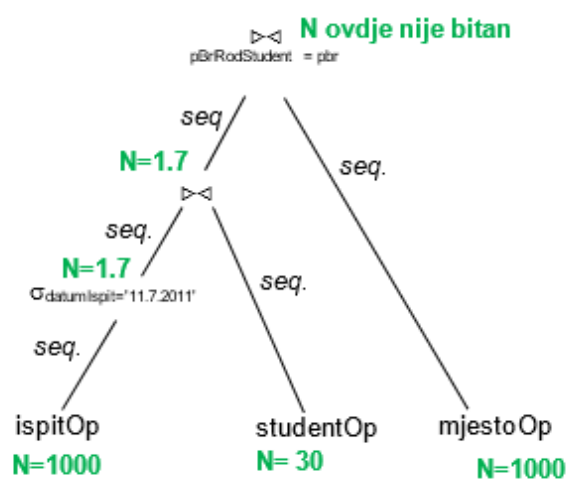
#### Određivanje redoslijeda spajanja relacija:

- $N(ispitOP2 \bowtie studentOp) = N(ispitOP2) * N(studentOP) / \max(V(JMBAG, ispitOP2), V(JMBAG, studentOP))$   
 $= 1.7 * 30 / \max(15, 30) = 1.7 * 30 / 30 = 1.7$   
Budući da nije kreiran primarni ključ u studentOP niti strani ključ u ispitOP, koristi se gornji izraz za procjenu broja n-torki.
- $N(ispitOP2 \bowtie mjestoOp) = N(ispitOP2) * N(mjestoOp) = 1.7 * 1000 = 1700$   
ispitOP i mjestoOP nemaju istoimenih atributa pa je prirodno spajanje zapravo Kartezijev produkt.
- $N(\underset{pbrRodStudent = pbr}{studentOp \bowtie mjestoOp}) = N(studentOP) * N(mjestoOP) / \max(V(pbrRodStudent, studentOP), V(pbr, mjestoOP)) = 30 * 1000 / \max(15, 1000) = 30 * 1000 / 1000 = 30$   
Kao kod procjenjivanja broja n-torki u rezultatu prirodnog spajanja ispitOp i studentOP, ni u ovom slučaju nije kreiran primarni ključ u mjestoOP niti strani ključ u studentOP, pa se koristi gornji izraz za procjenu broja n-torki.

Zaključak: redoslijed spajanja je:  $(ispitOP2 \bowtie studentOp) \bowtie mjestoOP$

Budući da ne postoji niti jedan indeks, jedina raspoloživa metoda pristupa je slijedno (sequential) čitanje blokova s podacima.

Feedback



## Primjer 2.

Prije obavljanja SELECT naredbe kreirajte sljedeće primarne (čime će biti kreirani i odgovarajući UNIQUE indeksi) i strane ključeve:

```
ALTER TABLE mjestoOp ADD CONSTRAINT mjestoOpPk PRIMARY KEY (pbr);
ALTER TABLE studentOp ADD CONSTRAINT pbrRodFk FOREIGN KEY (pBrRodStudent)
REFERENCES mjestoOp (pbr);
ALTER TABLE studentOp ADD CONSTRAINT studOpPk PRIMARY KEY (JMBAG);
ALTER TABLE ispitOp ADD CONSTRAINT JMBAGStudFk FOREIGN KEY (JMBAG)
REFERENCES studentOp (JMBAG);
```

```
SELECT *
FROM ispitOp, studentOp, mjestoOp
WHERE ispitOp.JMBAG = studentOp.JMBAG
AND studentOp.pBrRodStudent = mjestoOp.pbr
AND ispitOp.sifPred = 5013
AND mjestoOp.pbr = 51211;
```

rješenje

### Određivanje veličine međurezultata:

$studentOP2 = \sigma_{pbrRodStudent=51211}(studentOP)$

$N(studentOP2) = N(studentOP) / V(pbrRodStudent, studentOP) = 30 / 15 = 2$

$mjestoOP2 = \sigma_{pbrRodStudent=51211}(mjestoOP)$

$N(mjestoOP2) = N(mjestoOP) / V(pbr, mjestoOP) = 1000 / 1000 = 1$

Primijetite da se uvjet selekcije na poštanski broj može potisnuti i obaviti nad relacijom studentOP i nad relacijom mjestoOP.

$ispitOP2 = \sigma_{sifPred=5013}(ispitOP)$

$N(ispitOP2) = N(ispitOP10) / V(sifPred, ispitOP) = 1000 / 15 = 67$

### Određivanje redoslijeda spajanja relacija:

1.  $N(ispitOP2 \bowtie studentOp) = N(ispitOP2) = 67$

Budući da je kreiran primarni ključ u studentOP i strani ključ u ispitOP, sada SUBP zna da je presjek atributa shema relacija ispitOP2 i studentOP2 ključ relacije studentOP.

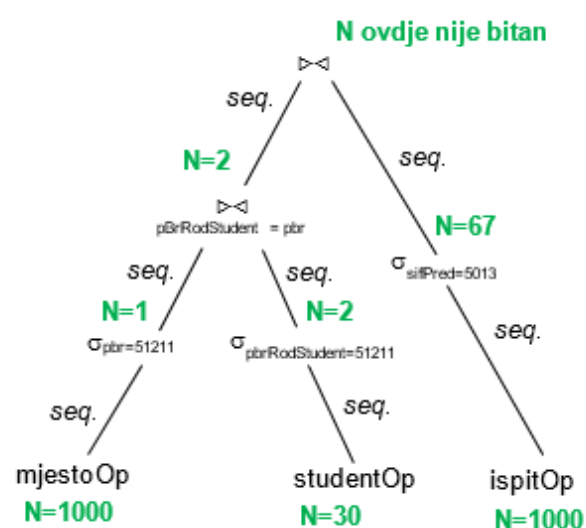
2.  $N(ispitOP2 \bowtie mjestoOP2) = N(ispitOP2) * N(mjestoOP2) = 67 * 1 = 67$

ispitOP i mjestoOP nemaju istoimenih atributa pa je prirodno spajanje Kartezijev produkt.

3.  $N(studentOp \bowtie_{pbrRodStudent = pbr} mjestoOp) = N(studentOP2) = 2$  Kao kod procjenjivanja broja n-torki u rezultatu prirodnog spajanja ispitOp i studentOP i u ovom slučaju SUBP zna za primarni ključ u mjestoOP i strani ključ u studentOP.

Zaključak: redoslijed spajanja je:  $N(studentOp \bowtie_{pbrRodStudent = pbr} mjestoOp) = \bowtie ispitOP2$

Budući da kao posljedica kreiranja mjestoOpPk postoji UNIQUE indeks za atribut pbr iz mjestoOP, uvjet selekcije nad n-torkama u mjestoOP može se obaviti čitanjem podataka pomoću indeksa. U svim ostalim slučajevima nema upotrebljivih indeksa i mora se koristiti slijedno (sequential) čitanje blokova s podacima.



### Primjer 3.

I dalje postoje primarni i strani ključevi kreirani u prethodnom primjeru. Postoje i odgovarajući indeksi - oni koji su nastali kao posljedica kreiranja primarnih ključeva.

```
SELECT *
FROM ispitOp, studentOp, mjestoOp
WHERE ispitOp.JMBAG = studentOp.JMBAG
AND studentOp.pBrRodStudent = mjestoOp.pbr
AND studentOp.prezimeStudent < 'Krivec';
```

rješenje

#### Određivanje veličine međurezultata:

$studentOP2 = \sigma_{\text{prezimeStudent} < 'Krivec'}(studentOP)$

$N(studentOP2) = N(studentOP) / 3 = 30 / 3 = 10$

#### Određivanje redoslijeda spajanja relacija:

1.  $N(ispitOp \bowtie studentOp2) = N(ispitOp) = 10000$

Svaka n-torka iz ispitOp će se spojiti s najviše jednom n-torkom iz studentOP2.

2.  $N(ispitOp \bowtie mjestoOp) = N(ispitOp) * N(mjestoOp) = 10000 * 1000 = 10000000$

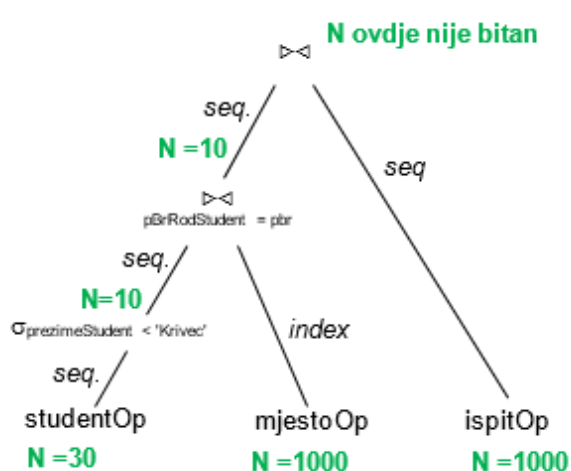
Prirodno spajanje je u ovom slučaju Kartezijev produkt.

3.  $N(studentOp \bowtie_{pBrRodStudent = pbr} mjestoOp) = N(studentOP2) = 10$  Kao kod procjenjivanja broja n-torki u rezultatu prirodnog spajanja ispitOp i studentOP, i u ovom slučaju SUBP zna za primarni ključ u mjestoOp i strani ključ u studentOP.

Zaključak: redoslijed spajanja je:  $N(studentOp \bowtie_{pBrRodStudent = pbr} mjestoOp) = \bowtie ispitOP2$

Prilikom spajanja relacija mjestoOp i studentOP2 (studentOP2 je međurezultat na kojem ne postoje indeksi) postoji mogućnost pristupa n-torkama relacije mjestoOp korištenjem UNIQUE indeksa kreiranog automatski prilikom kreiranja primarnog ključa mjestoOpPk.

Prilikom spajanja međurezultata dobivenog spajanjem relacija mjestoOp i studentOP2 s relacijom ispitOp jedina moguća metoda pristupa n-torkama relacije ispitOp je slijedno (sequential) jer ne postoji indeks nad JMBAG-om. Sustav PostgreSQL ne kreira automatski indeks kreiranjem stranog ključa JMBAGStudFk.

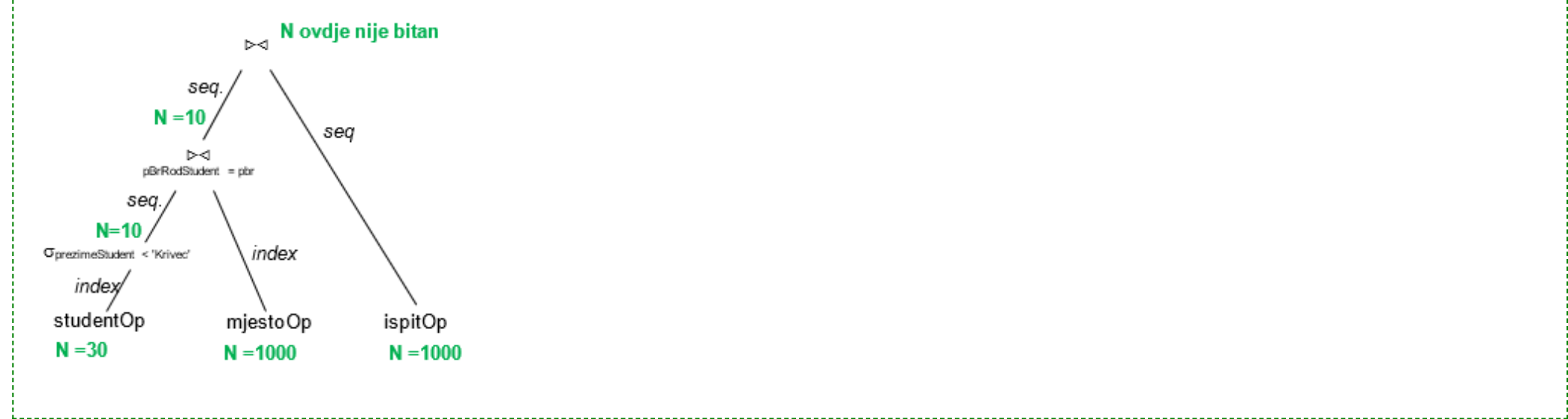


**Primjer 4.** Izvodi se SELECT naredba kao u Primjeru 3., ali je dodatno kreiran i indeks studPrezStud:

```
CREATE INDEX studPrezStud ON studentOp (prezimeStudent);
```

rješenje

Jedina razlika u odnosu na rješenje prethodnog primjera je način pristupa n-torkama iz studentOP pri obavljanju uvjeta selekcije.



Question

Playground

Ovaj zadatak se odnosi na bazu podataka studAdmin.  
Napišite SQL naredbe kojima ćete V(pbrRodStud, student), N(student) i N(ispit) ispisati u sljedećem obliku:

V	N
V (prezimeStudent, student)	477
V (sifPredmet, ispit)	44
V (pbrRodStudent, student)	41

Napišite sličan upit za tablice mjestoOP, studentOP i ispitOP. Uvjerite su da su istinite brojke dane za:  
V (pbrRodStudent, studentOP)  
V (prezimeStudent, studentOP)  
V (sifPredmet, ispitOP)  
V (datumIspit, ispitOP)

rješenje

```
SELECT 'V (pbrRodStudent, student)' V, count(DISTINCT pbrRodStudent) N FROM student
UNION
SELECT 'V (prezimeStudent, student)', count(DISTINCT prezimeStudent) FROM student
UNION
SELECT 'V (sifPredmet, ispit)', count(DISTINCT sifPredmet) FROM ispit
```

1

2

3

4

5

6

Run

Save

Feedback