



# Zadatci za vježbu

## #8

[Previous step](#)[Submit answer](#)[Next step](#)[Layout ▼](#)[Tutorial content ▼](#)

### 1/3: Privremene i virtualne relacije

#### Napomena vezana uz testiranje SQL naredbi iz rješenja.

SQL naredbe iz rješenja primjera u ovom i sljedećim koracima tutoriala možete izvoditi na dva načina:

1. kroz Edgara (npr. u Playgroundu)
2. u bazi podataka koju ste stvorili na vlastitoj instanci PostgreSQL sustava pomoću pgAdmina.

Preporučamo drugi način jer ćete u tom slučaju moći obavljati naredbu po naredbu i lakše ćete razumjeti postupak.

Također, u rješenjima primjera se podrazumijevaju da na taj način obavljate testiranje.

Ako ćete naredbe obavljati u okruženju Edgara vodite računa o tome da su efekti naredbi koje obavite prilikom jednog pokretanja (**Run**) prisutni samo za vrijeme njihovog izvršavanja. Nakon toga se baza podataka vraća u stanje u kojem je bila u trenutku neposredno prije pokretanja (obavljanja) naredbi.

Zbog toga je naredbu koja očekuje da su u bazi podataka sačuvani efekti neke druge, prethodne naredbe (ili više njih) potrebno obaviti zajedno sa svim tim naredbama.

Svi zadaci u ovom koraku se odnose na bazu podataka studAdmin.

#### Primjer 1.

Korisnik želi doznati podatke o studentima (jmbag, prezime i ime) koji su u akademskoj godini 2019./2020. ostvarili prosjek ocjena položenih predmeta veći ili jednak 4,1.

Budući da podatke planira trajno analizirati, korisnik ne želi stalno postavljati sljedeći SQL upit:

```
SELECT jmbag, prezimeStudent, imeStudent, AVG(ocjena)
FROM ispit
NATURAL JOIN student
WHERE (EXTRACT (YEAR FROM datumRok - '9 month'::INTERVAL)) = 2019
AND ocjena > 1
GROUP BY jmbag, prezimeStudent, imeStudent
HAVING AVG(ocjena)>=4.1;
```

Umjesto toga želi stvoriti:

a) privremenu relaciju tVrhunskiStudenti

b) virtualnu relaciju vVrhunskiStudenti

sa shemama:

TVRHUNSKISTUDENTI = {JMBAG, prezimeStudent, imeStudent, prosjek}

VVRHUNSKISTUDENTI = {JMBAG, prezimeStudent, imeStudent, prosjek}

Sadržaj privremene kao i virtualne relacije određen je gornjom SELECT naredbom.

Napišite SQL naredbe kojima će nastati shema i sadržaj relacije tVrhunskiStudenti, a potom i relacije vVrhunskiStudenti.

rješenje

a)

Da bismo stvorili privremenu relaciju tVrhunskiStudenti ispravne sheme, tj. da bismo ispravno odredili tipove podataka za attribute privremene relacije, provjerimo koji su tipovi podataka korišteni pri definiciji sheme relacije student i ispit.

```
CREATE TEMP TABLE tVrhunskiStudenti(
    JMBAG    VARCHAR(10)
    , prezimeStudent VARCHAR(25)
    , imeStudent   VARCHAR(25)
    , prosjek     DECIMAL(3,2));
```

Potom privremenu relaciju napunimo željenim podacima:

Feedback

```
INSERT INTO tVrhunskiStudenti (JMBAG, prezimeStudent, imeStudent, prosjek)
SELECT jmbag, prezimeStudent, imeStudent, AVG(ocjena)
FROM ispit
NATURAL JOIN student
WHERE (EXTRACT (YEAR FROM datumRok - '9 month'::INTERVAL)) = 2019
AND ocjena > 1
GROUP BY jmbag, prezimestudent, imestudent
HAVING AVG(ocjena)>=4.1;
```

b)  
Shema i sadržaj virtualne relacije vVrhunskiStudenti definirani su CREATE VIEW naredbom:

```
CREATE VIEW vVrhunskiStudenti (JMBAG, prezimeStudent, imeStudent, prosjek) AS
SELECT jmbag, prezimeStudent, imeStudent, AVG(ocjena)
FROM ispit
NATURAL JOIN student
WHERE (EXTRACT (YEAR FROM datumRok - '9 month'::INTERVAL)) = 2019
AND ocjena > 1
GROUP BY jmbag, prezimestudent, imestudent
HAVING AVG(ocjena)>=4.1;
```

Primjer 2.

Ispišimo podatke o studenatima i njihovom prosjeku, počevši od onog s najboljim prosjekom (ako više studenata ima jednak prosjek, poredajmo ih po prezimenu abecedno) i to

- a) pomoću relacije tVrhunskiStudenti,
- b) pomoću relacije vVrhunskiStudenti.
- c) Usporedimo rezultate u slučaju a) i slučaju b). Razlikuju li se?

rješenje

a)  
Temeljem sadržaja privremene relacije:

```
SELECT * FROM tVrhunskiStudenti
ORDER BY prosjek DESC, prezimeStudent;
```

U rezultatu je 56 n-torki:

JMBAG	prezimeStudent	prezimeStudent	prosjek
0555004806	Baljak	Ana	5.00
0555001708	Baričević	Boris	5.00
...	...	...	...
0555005359	Krmpotić	Eva	4.11
0555003125	Frančić	Smajil	4.10

b)  
Temeljem sadržaja virtualne relacije:

```
SELECT * FROM vVrhunskiStudenti
ORDER BY prosjek DESC, prezimeStudent;
```

U rezultatu je također 56 n-torki:

JMBAG	prezimeStudent	prezimeStudent	prosjek
0555004806	Baljak	Ana	5.00
0555001708	Baričević	Boris	5.00
...	...	...	...
0555005359	Krmpotić	Eva	4.11
0555003125	Frančić	Smajil	4.10

Feedback

c)  
Sadržaj privremene i virtualne relacije je jednak.

Primjer 3.

U međuvremenu se naknadno upisao student John Aiken i položio ispit iz predmeta „Fizika 1“ (šifra predmeta 34) dobivši ocjenu 5 (izvrstan). Dodajmo podatke o novom studentu, upisanom predmetu i položenom ispitu:

```
INSERT INTO student VALUES ('0036444444', 'John', 'Aiken', '19089463391', 'M', '19.08.1999', NULL, NULL);
INSERT INTO upisanpredmet VALUES ('0036444444', 34, 2019, 'A-A');
INSERT INTO ispit VALUES ('0036444444', 34, '03.02.2020', '05.02.2020', 5, 635);
```

Što se događa sa sadržajem privremene odnosno virtualne relacije nakon ovog ažuriranja?

rješenje

Pokrenimo ponovno upite

```
SELECT * FROM tVrhunskiStudenti
ORDER BY prosjek DESC, prezimeStudent;
```

U rezultatu je još uvijek 56 n-torki:

JMBAG	prezimeStudent	prezimeStudent	prosjek
0555004806	Baljak	Ana	5.00
0555001708	Baričević	Boris	5.00
...	...	...	...
0555005359	Krmpotić	Eva	4.11
0555003125	Frančić	Smajil	4.10
odnosno			

```
SELECT * FROM vVrhunskiStudenti
ORDER BY prosjek DESC, prezimeStudent;
```

U rezultatu je sada 1 n-torka više, odnosno 57 n-torki:

JMBAG	prezimeStudent	prezimeStudent	prosjek
00364444444	Aiken	John	5.00
0555004806	Baljak	Ana	5.00
0555001708	Baričević	Boris	5.00
...	...	...	...
0555005359	Krmpotić	Eva	4.11
0555003125	Frančić	Smajil	4.10

U slučaju privremene relacije tVrhunskiStudenti nema promjena u odnosu na trenutak kad smo prvi put definirali njen sadržaj. Privremena relacija ne registrira naknadne promjene nad sadržajem izvorne relacije.  
U virtualnu relaciju, naprotiv, „dodan“ je novi redak, budući da student John Aiken ima prosjek veći od 4,1.

Primjer 4.

U međuvremenu je studentu Frančić Smajilu (JMBAG 0555003125), zbog administrativne pogreške, ocjena iz predmeta Baze podataka smanjena s 4 na 3 (dobar):

```
UPDATE ispit SET ocjena = 3
WHERE jmbag = '0555003125'
AND sifPredmet = 34
AND ocjena = 4;
```

Provjerimo je li došlo do promjene u sadržaju virtualne relacije.

Feedback

rješenje

Izvođenjem upita

SELECT \* FROM vVrhunskiStudenti  
ORDER BY prosjek DESC, prezimeStudent;

dobije se sljedeći rezultat:

JMBAG	prezimeStudent	prezimeStudent	prosjek
0036444444	Aiken	John	5.00
0555004806	Baljak	Ana	5.00
0555001708	Baričević	Boris	5.00
...	...	...	...
0555005359	Krmpotić	Eva	4.11

Broj n-torki u relaciji vVrhunskiStudenti smanjio se za jedan.  
U rezultatu je sada ponovno 56 n-torki. I dalje je tu Aiken John, ali Frančić Smajila više nema jer je njegov prosjek sada manji od 4,1:

**Primjer 5.**

Prilikom rada s privremenom relacijom tVrhunskiStudenti odnosno virtualnom relacijom vVrhunskiStudenti klijentsko računalo je ostalo bez električne energije. To je potrajalo dulje vrijeme. Sustav za upravljanje bazom podataka nalazi se na zasebnom serveru (računalu) koje nije ostalo bez električne energije. Hoće li taj nestanak električne energije (i kako?) imati utjecaja na rad korisnika na klijentskom računalu?

rješenje

Nestankom električne energije klijentsko računalo se ugasio pa je time automatski došlo do prekida SQL-sjednice.

Prekidom SQL-sjednice

- Privremena relacija tVrhunskiStudenti je nestala. Korisnik mora nanovo stvoriti privremenu relaciju.
- Virtualna relacija vVrhunskiStudenti i dalje postoji.

Question

Playground

Napraviti virtualnu relaciju **boravakVikendom** sa shemom **BORAVAKVIKENDOM = JMBAG, ukUlazaka** koja će omogućiti pregled ulazaka u dvorane vikendom za sve studenate (*evidencijaBoravka.datumVrijemePoc*).

Primjer rezultata:

SELECT \* FROM boravakVikendom;

JMBAG	ukUlazaka
0555004991	0
0555003312	26
0555003008	0
0555002908	0
0555000828	26
...	...

Primijetite da se za studente koji nisu ulazili u dvorane vikendom za ukupan broj ulazaka u dvorane ispisuje vrijednost 0.

rješenje

CREATE VIEW boravakVikendom (JMBAG, ukUlazaka) AS

Feedback

```
SELECT student.JMBAG, COUNT(datumVrijemePoc)
FROM student
LEFT JOIN evidencijaBoravka
      ON student.jmbag = evidencijaBoravka.jmbag
      AND EXTRACT(DOW FROM datumVrijemePoc) IN (0,6)
GROUP BY student.JMBAG;
```

1

2

3

4

5

6

Run

Save