



Zadatci za vježbu

#8

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

2/3: Migrirajuće n-torke

Primjeri u ovom koraku tutoriala također se odnose na bazu podataka studAdmin. Također, u rješenjima primjera podrazumijeva se da testiranje obavljate u bazi podataka studAdmin koju ste stvorili na vlastitoj instanci PostgreSQL sustava pomoću pgAdmina.

Primjer 1.

Želimo promatrati koliko su ukupno ECTS bodova ostvarili studenti slušanjem svih predmeta od početka svog studija do danas.

Načinimo dvije virtualne relacije čija relacijska shema sadrži sljedeće attribute:

{JMBAG, prezimeStudent, imeStudent, ukupnoBodova}.

Atribut ukupnoBodova je suma svih ECTS bodova koje je student osvojio uspješnim polaganjem predmeta tijekom svog studiranja, od početka do danas.

Potrebno je načiniti sljedeće dvije virtualne relacije čije se sheme kao što je zadano:

- vOsvojiliManjeOd25, sadrži studente čija je suma ECTS bodova manja od 25.
- vOsvojiliOd25Do50, sadrži studente čija je suma ECTS bodova veća ili jednaka 25 i manja ili jednaka 50.

Ispišite sadržaj ovih virtualnih relacija tako da zapisi budu poredani prema broju osvojenih ECTS- a silazno, a potom prema prezimenu uzlazno.

Provjerite nad kojom virtualnom relacijom treba obaviti upit da se dobiju podaci za studenta s JMBAG-om '0555004010'.

[rješenje](#)

Sljedećim SQL naredbama se mogu stvoriti zadane virtualne relacije:

```
CREATE VIEW vOsvojiliManjeOd25 (JMBAG, prezimeStudent, imeStudent, ukupnoBodova) AS
SELECT jmbag, prezimestudent, imeStudent, SUM(ectsbod)
FROM predmet
NATURAL JOIN ispit
NATURAL JOIN student
WHERE ocjena > 1
GROUP BY jmbag, prezimestudent, imestudent
HAVING SUM(ectsbod)<25;
```

```
i
CREATE VIEW vOsvojiliOd25Do50 (JMBAG, prezimeStudent, imeStudent, ukupnoBodova) AS
SELECT jmbag, prezimestudent, imeStudent, SUM(ectsbod)
FROM predmet
NATURAL JOIN ispit
NATURAL JOIN student
WHERE ocjena > 1
GROUP BY jmbag, prezimestudent, imestudent
HAVING SUM(ectsbod) BETWEEN 25 AND 50;
```

Izvršavanjem naredbi:

```
SELECT * FROM vOsvojiliManjeOd25 ORDER BY ukupnoBodova DESC, prezimeStudent;
SELECT * FROM vOsvojiliOd25Do50 ORDER BY ukupnoBodova DESC, prezimeStudent;
```

Ispisuje se 160 zapisa virtualne relacije vOsvojiliManjeOd25 i 82 zapisa virtualne relacije vOsvojiliOd25Do50 .

Izvršavanjem upita:

```
SELECT * FROM vOsvojiliManjeOd25 WHERE JMBAG = '0555004010';
```

dobije se sljedeći rezultat:

[Feedback](#)

JMBAG	prezimeStudent	imeSTudent	ukupnoBodova
0555004010	Hibšer	Krešimir	23.0

Upit

```
SELECT * FROM vOsvojiliOd25Do50 WHERE JMBAG = '0555004010';
```

ne treba niti pokretati jer znamo da rezultat sigurno neće sadržavati niti jednu n-torku.

Primjer 2.

U međuvremenu je student Krešimir Hibšer (JMBAG 0555004010) položio ocjenom 5 predmet 'Informacijske mreže' (sifPredmet=30) na ispitnom roku s datumom 01.02.2021. Evidentirajmo ovaj događaj izvršavanjem upita:

```
UPDATE ispit SET ocjena = 5
WHERE JMBAG = '0555004010'
AND sifPredmet = 30
AND datumRok = '01.02.2021';
```

Provjerimo nad kojom virtualnom relacijom sada treba obaviti upit da se dobiju podaci za studenta s JMBAG-om '0555004010'

rješenje

Student Krešimir Hibšer je prije polaganja predmeta 'Informacijske mreže' ukupno osvojio 23 ECTS-a. Budući da je broj ECTS-a predmeta 'Informacijske mreže' jednak 6, nakon položenog ispita, student će imati 29 ECTS-a. Zbog toga se njegovi podaci više neće nalaziti u virtualnoj relaciji vOsvojiliManjeOd25 nego u vOsvojiliOd25Do50.

vOsvojiliManjeOd25 sada ima ukupno 159 zapisa umjesto ranijih 160, a vOsvojiliOd25Do50 83 umjesto ranijih 82.

Izvršavanjem upita:

```
SELECT * FROM vOsvojiliManjeOd25 WHERE JMBAG = '0555004010';
```

ne dobije se niti jedna n-torka. Izvršavanjem upita:

```
SELECT * FROM vOsvojiliOd25Do50 WHERE JMBAG = '0555004010';
```

dobije se sljedeći rezultat:

JMBAG	prezimeStudent	imeSTudent	ukupnoBodova
0555004010	Hibšer	Krešimir	29.0

Primjer 3.

Pokušajmo, korištenjem virtualne relacije vOsvojiliManjeOd25, studentima koji su osvojili ukupno 24 ECTS boda promijeniti prezime u vrijednost 'Nepoznato'.

rješenje

Izvedemo li naredbu

```
UPDATE vOsvojiliManjeOd25
SET prezimeStudent = 'Nepoznato' WHERE ukupnobodova=24;
```

SUBP će javiti grešku:

```
ERROR: cannot update view "vosvojilimanjeod25"
DETAIL: Views containing GROUP BY are not automatically updatable.
HINT: To enable updating the view, provide an INSTEAD OF UPDATE trigger or an unconditional ON UPDATE DO INSTEAD rule.
SQL state: 55000
```

Zašto?

Virtualna relacija vOsvojiliManjeOd25 je neizmjenjiva.

Isto vrijedi i za virtualnu relaciju vOsvojiliOd25Do50.

Sve virtualne relacije koje u glavnom SELECT dijelu svoje definicije koriste atribut više relacija su neizmjenjive.

Također, neizmjenjive su i virtualne relacije koje u glavnom dijelu SELECT naredbe koriste atribut samo jedne relacije, ali pri tom sadrže grupiranje (GROUP BY).

U definiciji obje virtualne relacije koristi se grupiranje.

Definicije virtualnih relacija vOsvojiliManjeOd25 i vOsvojiliOd25Do50 nije moguće napisati tako da budu izmjenjive zbog toga što sadrže izraz ukupnoBodova koji se dobije agregiranjem.

Prezime studenata koji su osvojili 24 ECTS-a ukupno može se postaviti na 'Nepoznato' sljedećom naredbom:

```
UPDATE student
  SET prezimeStudent='Nepoznato'
 WHERE jmbag IN (
   SELECT jmbag
   FROM predmet
   NATURAL JOIN ispit
   NATURAL JOIN student
   WHERE ocjena > 1
   GROUP BY jmbag, prezimestudent, imestudent
   HAVING SUM(ectsbod)=24);
```

Primjer 4.

Načinimo virtualnu relaciju vPredmetiSBarem7ECTS pomoću koje ćemo moći pregledavati podatke o predmetima (šifra, naziv predmeta i broj ECTS bodova), sa 7 ili više ECTS bodova. Pri tom možemo računati da virtualnu relaciju nećemo koristiti za ažuriranje podataka o predmetima.

Zanemarimo sada prethodnu rečenicu pa pokušajmo dodati novi predmet (sifra=111, naziv=“Novi“, ects=4) korištenjem virtualne relacije vPredmetiSBarem7ECTS. Ispišimo sadržaj relacije predmet za sve šifre veće od 100.

rješenje

Virtualna relacija vPredmetiSBarem7ECTS stvara se naredbom

```
CREATE VIEW vPredmetiSBarem7ECTS (sifpredmet, nazpredmet, ectsbod) AS
SELECT sifpredmet, nazpredmet, ectsbod
  FROM predmet
 WHERE predmet.ectsbod>=7;
```

Sadržaj virtualne relacije (13 n-torki) moguće je procjeriti pomoću sljedeće naredbe:

```
SELECT * FROM vPredmetiSBarem7ECTS;
```

Pokuša li se izvršiti sljedeća naredba:

```
INSERT INTO vPredmetiSBarem7ECTS VALUES (111, 'Novi', 4);
```

SUBP javlja uspješan ishod izvršavanja. INSERT 0 1 Query returned successfully in 48 msec.

Međutim, sadržaj virtualne relacije vPredmetiSBarem7ECTS se ne mijenja: U rezultatu naredbe

```
SELECT * FROM vPredmetiSBarem7ECTS;
```

je i dalje 13 zapisa.

Logično, budući da virtualna relacija sadrži samo predmete sa 7 i više ECTS bodova.

Ako pogledamo sadržaj temeljne relacije predmet (za sve zapise sa šifrom većom od 100) vidimo da je naredba INSERT preko virtualne relacije vPredmetiSBarem7ECTS stvorila novu n-torku u relaciji predmet.

```
SELECT *
FROM predmet
WHERE sifPredmet>100;
```

Rezultat naredbe je:

Feedback

sifpredmet	nazpredmet	ectsbod	ukbrsatitjedno
111	Novi	4	(null)

Vrijednost atributa ukbrsatitjedno za danu n-torku je nepoznata - null.

To je očekivano jer u definiciji virtualne relacija vPredmetiSBarem7ECTS nije korišten atribut ukbrsatitjedno pa stoga niti ne omogućava postavljanje vrijednosti tog atributa.

Da je taj atribut definiran s NOT NULL ograničenjem gornju insert naredbu ne bi bilo moguće uspješno obaviti.

Obrišimo predmet sa šifrom 111:

```
DELETE FROM predmet WHERE sifPredmet = 111;
```

Primjer 5.

Načinimo virtualnu relaciju v2PredmetiSBarem7ECTS sličnu relaciji vPredmetiSBarem7ECTS iz prethodnog zadatka (omogućava pregled podataka o predmetima s barem 7 ECTS-a), ali ovaj put uzmimo u obzir da ćemo v2PredmetiSBarem7ECTS koristiti i za ažuriranje podataka o predmetima s barem 7 ECTS-a.

Pokušajmo dodati novi predmet (sifra=111, naziv=“Novi“, ects=4) kroz virtualnu relaciju v2PredmetiSBarem7ECTS. Pokušajmo isto učiniti za predmet (sifra=112, naziv=“Najnoviji“, ects=12).

rješenje

Virtualna relacija v2PredmetiSBarem7ECTS koja će se, između ostaloga, koristiti i za ažuriranje n-torki u predmet stvara se naredbom:

```
CREATE VIEW v2PredmetiSBarem7ECTS (sifPredmet, nazPredmet, ectsBod) AS
SELECT sifpredmet, nazpredmet, ectsbod
  FROM predmet
 WHERE predmet.ectsbod>=7
WITH CHECK OPTION;
```

Izraz WITH CHECK OPTION **onemogućuje** da se u virtualnu relaciju v2PredmetiSBarem7ECTS dodaju/mijenjaju/brišu n-torke koje ne zadovoljavaju njenu definiciju (one koje se kasnije neće moći pregledati pomoću te virtualne relacije).

Sadržaj virtualne relacije v2PredmetiSBarem7ECTS jednak je sadržaju virtualne relacije vPredmetiSBarem7ECTS.

Izvršavanje naredbe:

```
INSERT INTO v2PredmetiSBarem7ECTS VALUES (111, 'Novi', 4);
```

završi pogreškom:

```
ERROR:  new row violates check option for view "v2predmetisbarem7ects"
DETAIL:  Failing row contains (111, Novi, 4.0, null).
SQL state: 44000
```

Ta n-torka ne zadovoljava definiciju relacije v2PredmetiSBarem7ECTS i ne bi se mogla pregledati obavljanjem naredbe

```
SELECT * FROM v2PredmetiSBarem7ECTS;
```

Naredba

```
INSERT INTO v2PredmetiSBarem7ECTS VALUES (112, 'Najnoviji', 12);
```

se uspješno izvede, a u relaciji v2PredmetiSBarem7ECTS pojavljuje se nova n-torka.

```
SELECT * FROM v2PredmetiSBarem7ECTS WHERE sifPredmet > 100;
```

sifpredmet	nazpredmet	ectsbod	ukbrsatitjedno
111	Najnoviji	12	(null)

Napraviti izmjenjivu virtualnu relaciju **samoPeticeU2020** sa shemom jednakom shemi relacije ispit koja će se moći koristiti za evidenciju ispita položenih s ocjenom 5 na ispitnim rokovima s datumom ispitnog roka u 2020 kalendarskoj godini.

rješenje

```
CREATE VIEW samopeticeU2020 (JMBAG, sifpredmet, datumrok, datumispit, ocjena, sifnastavnik) AS
SELECT *
FROM ispit
WHERE ocjena = 5
AND EXTRACT (YEAR FROM datumRok) = 2020
WITH CHECK OPTION;
```

1

2

3

4

5

6

Run

Save