



Zadatci za vježbu

#11

1/2

: Pohranjene funkcije i procedure

Primjere treba testirati u bazi **studAdmin** koju ste stvorili na vlastitoj instanci *PostgreSQL* sustava (npr. pomoću *pgAdmina*).

Primjer 1.

Napišimo funkciju koja će za zadani JMBAG studenta vratiti ukupni broj upisanih ECTS bodova (cijeli broj) za tog studenta.

rješenje

```
CREATE FUNCTION getEcts (p_JMBAG VARCHAR(10)) RETURNS INTEGER
AS $$
BEGIN
    RETURN(SELECT  SUM(ectsbod)
            FROM  predmet JOIN upisanpredmet
            ON  predmet.sifpredmet = upisanpredmet.sifpredmet
            WHERE upisanpredmet.jmbag = p_JMBAG);
END;
$$ LANGUAGE plpgsql;
```

Napomena: ako ćete nakon kreiranja trebati raditi izmjene funkcije, koristite CREATE OR REPLACE FUNCTION...

Funkciju sad možemo iskoristiti, na primjer, da za sve studente koji imaju barem jedan upisani predmet ispišemo JMBAG, ime, prezime i ukupni broj upisanih ECTS bodova.

```
select JMBAG, imeStudent, prezimeStudent, getEcts (jmbag) AS ECTSukupno
from student
where getEcts(jmbag) IS NOT NULL;
```

Sjetimo se da je rezultat operacije SUM nad praznim skupom vrijednosti NULL.

Izvođenjem upita dobija se sljedeći rezultat:

JMBAG	prezimeStudent	prezimeStudent	ECTSukupno
0555000950	Mijo	Klasić	79
0555001092	Josip	Janeš	70
...

Primjer 2.

Napišimo pohranjenu funkciju koja će za zadani datum vratiti akademsku godinu kojoj taj datum pripada. Pretpostavite da akademska godina traje od 1.10. jedne, do 30.9. sljedeće kalendarske godine. Npr. za datum 23.7.2021., akademska godina je 2020/2021 i procedura treba vratiti **2020**. Za datum 3.10.2019. procedura treba vratiti **2019**.

rješenje

```
CREATE OR REPLACE FUNCTION getAkGodina(p_date DATE) RETURNS INTEGER AS $$  
BEGIN  
    IF (EXTRACT( MONTH FROM p_date) >= 10) THEN  
        RETURN EXTRACT( YEAR FROM p_date );  
    ELSE  
        RETURN EXTRACT( YEAR FROM p_date )-1;  
    END IF;  
END;  
$$ LANGUAGE plpgsql;
```

Funkciju možemo pozvati, na primjer, u sljedećoj SELECT naredbi:

```
select jmbag, datumRok, getAkGodina (datumRok) AS akGod, ocjena  
from ispit  
where ocjena = 5;
```

Kao rezultat izvođenja te naredbe dobivamo:

JMBAG	datumRok	akGod	ocjena
0555000490	2020-06-13	2019	5
0555000490	2019-02-17	2018	5
...

Primjer 3.

OIB ima 11 znamenaka, a posljednja tj. 11. znamenka je kontrolna znamenka koja je dobivena izračunom iz prethodnih 10 znamenaka sljedećim postupkom:

1. prva znamenka zbroji se s brojem 10
2. računamo ostatak cjelobrojnog dijeljenja zbroj iz 1. koraka s 10; ako je dobiveni ostatak 0, on se zamijeni brojem 10
3. dobiveni ostatak pomnoži se brojem 2
4. računamo ostata cjelobrojnog dijeljenja dobivenog umnoška s 11; ovaj ostatak matematički nikako ne može biti 0 jer je rezultat prethodnog koraka uvijek paran broj
5. sljedeća znamenka zbroji se s ostatkom u prethodnom koraku
6. ponavljaju se koraci 2, 3, 4 i 5 dok se ne potroše sve znamenke
7. razlika između broja 11 i ostatka u zadnjem koraku je kontrolna znamenka; ako je ostatak 1 kontrolna znamenka je 0 (11-1=10, a 10 ima dvije znamenke)

Napišimo pohranjenu funkciju koja će za zadani OIB vratiti 1 ako je OIB ispravan, odnosno 0 ako nije.

rješenje

```
CREATE OR REPLACE FUNCTION checkOIB (p_oib VARCHAR(11))
  RETURNS SMALLINT AS $$

DECLARE
  kz    SMALLINT;
  broj  SMALLINT;
  i     SMALLINT;

BEGIN
  IF (CHAR_LENGTH(p_oib) != 11) THEN
    RETURN 0;
  END IF;

  IF (SUBSTRING(p_oib FROM 11 FOR 1) BETWEEN '0' AND '9') THEN
    FOR i IN 1..10 LOOP
      IF i = 1 THEN
        broj = SUBSTRING(p_oib FROM i FOR 1)::INTEGER + 10;
      ELSE
        broj = SUBSTRING(p_oib FROM i FOR 1)::INTEGER + broj;
      END IF;

      broj = broj % 10;

      IF broj = 0 THEN
        broj = 10;
      END IF;

      broj = broj * 2;
      broj = broj % 11;
    END LOOP;
    ELSE RETURN 0;
  END IF;

  IF broj = 1 THEN
    kz = 0;
  ELSE
    kz = 11 - broj;
  END IF;

  IF ((SUBSTRING(p_oib FROM 11 FOR 1)::INTEGER) != kz) THEN
    RETURN 0;
  END IF;

  RETURN 1;
END;
$$ LANGUAGE plpgsql;
```

Ako u SELECT naredbi pozovemo funkciju za OIB s vrijednošću 69435151530:

```
SELECT checkOIB ('69435151530') AS provjeraOIB;
```

dobit ćemo:

provjeraOIB
1

Za OIB 69435151531 imat ćemo:

```
SELECT checkOIB ('69435151531') AS provjeraOIB;
```

provjeraOIB
0

Napišimo pohranjenu proceduru koja će za zadani JMBAG, šifru predmeta i oznaku grupe obaviti "upis" predmeta tj. dodavanje zapisa u tablicu **upisanPredmet**, pod uvjetom da student već nema upisan taj predmet u tekućoj akademskoj godini. Drugi uvjet je da je za taj predmet već definirana ta grupa u tablici **predmetgrupa** i to za tekuću akademsku godinu. Ako to nije slučaj, treba ispisati poruku npr.:

Nema grupe A-M za predmet 32 u tekućoj ak.god.

i prekinuti s radom. Procedura vraća 1 ako je predmet uspješno upisan, odnosno 0 ako se ne može upisati (jer student već ima upisan taj predmet u tekućoj akademskoj godini). Za određivanje tekuće akademske godine iskoristite proceduru iz primjera 2.

rješenje

```
CREATE OR REPLACE FUNCTION upisPredmeta (p_JMBAG VARCHAR(10), p_sifPredmet INTEGER, p_oznGrupa VARCHAR (10))
  RETURNS SMALLINT AS $$

BEGIN
  IF NOT EXISTS (SELECT *
                 FROM predmetgrupa
                 WHERE sifPredmet = p_sifPredmet
                 AND akgodina = getAkGodina(CURRENT_DATE)
                 AND oznGrupa = p_oznGrupa)
  THEN
    RAISE EXCEPTION 'Nema grupe % za predmet % u tekućoj ak.god.', p_oznGrupa, p_sifPredmet;
  END IF;

  IF ( NOT EXISTS ( SELECT *
                   FROM upisanPredmet
                   WHERE jmbag = p_JMBAG
                   AND sifPredmet = p_sifPredmet
                   )
  )
  THEN
    INSERT INTO upisanPredmet
    VALUES (p_JMBAG, p_sifPredmet, getAkGodina(CURRENT_DATE), p_oznGrupa);
    RETURN 1;
  END IF;

  RETURN 0;
END;
$$ LANGUAGE plpgsql;
```

Testirajte proceduru korištenjem podataka iz baze **studAdmin**.

Primjer poziva funkcije:

```
SELECT upisPredmeta ('0555000529', 32, 'A-B');
```

[Question](#)

[Playground](#)

Koja od navedenih SELECT naredbi, u kojima se poziva pohranjena funkcija **checkOIB** iz primjera 2, **nije** ispravna?

a SELECT checkOIB (oib) FROM student WHERE imeStudent = 'Vedran';

b SELECT checkOIB (69435151530);

Feedback

c

SELECT * FROM student WHERE checkOIB (oib) = 1;

d

SELECT * FROM student WHERE checkOIB (oib) > 0;

Edgar: On-Line Exam Web Application © Developed with ❤ @FER