



# Zadatci za vježbu

## #10

Previous step

Submit answer

Next step

Layout ▾

Tutorial content ▾

3/4

## : Dodjeljivanje dozvola na razini baze podataka, na razini sheme, za objekte unutar sheme

Pretpostavka je da su zapamćeni efekti naredbi koje je korisnik *postgres*, administrator baze podataka **studAdmin**, obavio u prethodnom koraku:

```
REVOKE CONNECT ON DATABASE "studAdmin" FROM public;
REVOKE ALL ON SCHEMA public FROM public;
REVOKE ALL ON ALL TABLES IN SCHEMA public FROM public;
```

### Primjer 1.

Kao korisnik *postgres* spojiti se na sustav **PostgreSQL 13**.

- Kreirati uloge *horvat*, *kolar* i *novak* s mogućnosti uspostavljanja SQL sjednice (koristiti autorizaciju pomoću lozinke). Korisnicima za inicijalnu lozinku postaviti ime korisnika, npr. “horvat”.
- Korisnicima *horvat*, *kolar* i *novak* omogućiti pristup bazi podataka bez mogućnosti kreiranja novih objekata, te mogućnost pregledavanja/korištenja objekata sheme *public*.
- Svim korisnicima baze podataka omogućiti pregled podataka u relacijama *mjesto* i *zupanija*.
- Korisniku *novak* dodijeliti dozvolu za stvaranje shema na bazi podataka **studAdmin**. Kao korisnik *novak* napraviti novu shemu *nastava*.
- Korisniku *novak* omogućiti pregled, unos, izmjenu i brisanje svih podataka o predmetima (relacija *predmet*) s mogućnosti dodjeljivanja tih dozvola drugim korisnicima. Kao korisnik *novak* dodijeliti dozvolu pregledavanja podataka o predmetima korisniku *horvat*.
- Korisniku *novak* dodijeliti dozvolu za pregled samo OIB-a, imena i prezimena svih studenata evidentiranih u relaciji *student*.
- Korisniku *novak* dodijeliti dozvolu za stvaranje vlastitih objekata u shemi *public*. Kao korisnik *novak* napraviti nove relacije: *ocjena* (*ocjena* SMALLINT PRIMARY KEY, *opisOcjena* VARCHAR(10) NOT NULL) u shemi *public*, te *ocjenaNastavnikPredmet* (*sifNastavnik* INTEGER, *sifPredmet* INTEGER, *JMBAG* VARCHAR(10), *ocjena* SMALLINT NOT NULL, PRIMARY KEY(*sifNastavnik*, *sifPredmet*, *JMBAG*)) u shemi *nastava*.
- Korisnicima *horvat*, *kolar* i *novak* oduzeti sve dozvole koje su dobili u zadacima od 2. do 6.

### rješenje

1.

--postgres:

```
CREATE ROLE horvat LOGIN PASSWORD 'horvat'; --ROLE i USER su sinonimi
CREATE ROLE kolar  LOGIN PASSWORD 'kolar';
CREATE ROLE novak  LOGIN PASSWORD 'novak';
```

2.

--postgres:

```
GRANT CONNECT ON DATABASE "studAdmin" TO horvat, kolar, novak;
GRANT USAGE ON SCHEMA PUBLIC TO horvat, kolar, novak;
SELECT * FROM pg_user;
```

Sada se korisnici *horvat*, *kolar* i *novak* mogu spojiti na bazu podataka. Uvjerite se da je to doista tako - uspostavite korisničku sjednicu s korisničkim imenom *novak*. U prvom koraku tutoriala je opisano kako to možete napraviti.

3.

--postgres:

```
GRANT SELECT ON mjesto, zupanija TO PUBLIC;
```

Kao korisnik *novak* obavite sljedeće SQL naredbe:

--novak:

Feedback

```
SELECT * FROM mjesto;  
SELECT * FROM zupanija;
```

4.

--postgres:

```
GRANT CREATE ON DATABASE "studAdmin" TO novak;
```

--novak

```
CREATE SCHEMA nastava;
```

5.

--postgres:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON predmet TO novak WITH GRANT OPTION;
```

--novak:

```
INSERT INTO predmet VALUES (67, 'Test predmet', 1.0, 1);  
SELECT * FROM predmet WHERE sifPredmet = 67;  
  
UPDATE predmet SET ectsBod = 2.0, ukBrSatiTjedno = 2 WHERE sifPredmet = 67;  
SELECT * FROM predmet WHERE sifPredmet = 67;  
  
DELETE FROM predmet WHERE sifPredmet = 67;  
SELECT * FROM predmet WHERE sifPredmet = 67;  
  
GRANT SELECT ON predmet TO horvat;
```

--horvat:

```
SELECT * FROM predmet;
```

6.

--postgres:

```
GRANT SELECT (OIB, imeStudent, prezimeStudent) ON student TO novak;
```

--novak:

```
SELECT OIB, imeStudent, prezimeStudent FROM student;
```

7.

--postgres:

```
GRANT CREATE ON SCHEMA PUBLIC TO novak;
```

--novak:

```
CREATE TABLE public.ocjena (  
    ocjena SMALLINT PRIMARY KEY,  
    opisOcjena VARCHAR (10) NOT NULL  
);  
  
CREATE TABLE nastava.ocjenaNastavnikPredmet (  
    sifNastavnik INTEGER,  
    sifPredmet INTEGER,  
    JMBAG CHAR (10),  
    ocjena SMALLINT,  
    PRIMARY KEY (sifNastavnik, sifPredmet, JMBAG)  
);  
  
INSERT INTO public.ocjena VALUES  
    (1, 'nedovoljan'),  
    (2, 'dovoljan'),  
    (3, 'dobar'),  
    (4, 'vrlo dobar'),  
    (5, 'odličan');  
  
INSERT INTO nastava.ocjenaNastavnikPredmet VALUES  
    (33, 720, 0555003989, 4),  
    (44, 656, 0555003648, 2);  
  
SELECT ONP.sifNastavnik, ONP.sifPredmet, O.opisOcjena  
FROM public.ocjena AS O  
    INNER JOIN nastava.ocjenaNastavnikPredmet AS ONP  
        ON O.ocjena = ONP.ocjena;
```

8.

--postgres:

```
REVOKE SELECT ON mjesto, zupanija FROM PUBLIC;  
REVOKE CREATE ON DATABASE "studAdmin" FROM novak;  
REVOKE SELECT, INSERT, UPDATE, DELETE ON predmet FROM novak CASCADE;  
REVOKE SELECT (OIB, imeStudent, prezimeStudent) ON student FROM novak;  
REVOKE CREATE ON SCHEMA PUBLIC FROM novak;
```

[Question](#)

[Playground](#)

U zadatku pod 5. korisnik *novak* je korisniku *horvat* omogućio pregled podataka u tablici predmet s naredbom GRANT SELECT ON predmet TO horvat;

Što se dogodilo s tom dozvolom?

**a**

*horvat* još uvijek ima dozvolu pregledavanja podataka u predmet.

**b**

Ona je ukinuta naredbom:

```
REVOKE SELECT, INSERT, UPDATE, DELETE ON predmet FROM novak CASCADE;
```

a važnu ulogu je odigrala opcija CASCADE. Odnosno, ukidanjem dozvole korisniku *novak* kaskadno je ukinuta i dozvola korisniku *horvat*.