

Zadatak 1

- baza *studAdmin*
- Korisnik *postgres* je administrator baze podataka te nitko osim njega nema dozvolu pristupa bazi podataka niti objektima baze podataka.
- Pretpostaviti kako su nakon kreiranja *studAdmin* baze podataka pokrenute sljedeće naredbe koje ukidaju preddefinirane ovlasti za korisnike na *public* shemi:

REVOKE CONNECT ON DATABASE studAdmin FROM public;

REVOKE ALL ON SCHEMA public FROM public;

REVOKE ALL ON ALL TABLES IN SCHEMA public FROM public;

a) Kreirati korisnika *kolar* s mogućnošću uspostavljanja SQL sjednice, koristiti autorizaciju pomoću lozinke. Korisniku za inicijalnu lozinku postaviti ime korisnika (*kolar*).

CREATE USER _____

Zadatak

- baza *studAdmin*
- Korisnik *postgres* je administrator baze podataka te nitko osim njega nema dozvolu pristupa bazi podataka niti objektima baze podataka.
- Pretpostaviti kako su nakon kreiranja *studAdmin* baze podataka pokrenute sljedeće naredbe koje ukidaju preddefinirane ovlasti za korisnike na *public* shemi:

REVOKE CONNECT ON DATABASE studAdmin FROM public;

REVOKE ALL ON SCHEMA public FROM public;

REVOKE ALL ON ALL TABLES IN SCHEMA public FROM public;

a) Kreirati korisnika *kolar* s mogućnošću uspostavljanja SQL sjednice, koristiti autorizaciju pomoću lozinke. Korisniku za inicijalnu lozinku postaviti ime korisnika (*kolar*).

CREATE USER kolar WITH PASSWORD 'kolar';

Zadatak

- baza *studAdmin*
- Korisnik *postgres* je administrator baze podataka te nitko osim njega nema dozvolu pristupa bazi podataka niti objektima baze podataka.
- Pretpostaviti kako su nakon kreiranja *studAdmin* baze podataka pokrenute sljedeće naredbe koje ukidaju preddefinirane ovlasti za korisnike na *public* shemi:

REVOKE CONNECT ON DATABASE studAdmin FROM public;

REVOKE ALL ON SCHEMA public FROM public;

REVOKE ALL ON ALL TABLES IN SCHEMA public FROM public;

b) Korisniku *kolar* omogućiti pristup bazi podataka bez mogućnosti kreiranja novih objekata, te mogućnost pregledavanja objekata sheme *public*.

GRANT _____ **ON** _____ **"studAdmin" TO kolar;**

GRANT _____ **ON** _____ **PUBLIC TO kolar;**

Zadatak

- baza *studAdmin*
- Korisnik *postgres* je administrator baze podataka te nitko osim njega nema dozvolu pristupa bazi podataka niti objektima baze podataka.
- Pretpostaviti kako su nakon kreiranja *studAdmin* baze podataka pokrenute sljedeće naredbe koje ukidaju preddefinirane ovlasti za korisnike na *public* shemi:

REVOKE CONNECT ON DATABASE studAdmin FROM public;

REVOKE ALL ON SCHEMA public FROM public;

REVOKE ALL ON ALL TABLES IN SCHEMA public FROM public;

b) Korisniku *kolar* omogućiti pristup bazi podataka bez mogućnosti kreiranja novih objekata, te mogućnost pregledavanja objekata sheme *public*.

GRANT CONNECT ON DATABASE studAdmin TO kolar;

GRANT USAGE ON SCHEMA PUBLIC TO kolar;

Zadatak

c) Korisniku *kolar* dodijeliti dozvolu za stvaranje shemi na bazi podataka *studAdmin*. Kao korisnik *kolar* napraviti novu shemu (SCHEMA) *nastava*.

--postgres:

GRANT _____ ON DATABASE studAdmin TO kolar;

--kolar

CREATE _____ ;

Zadatak

c) Korisniku *kolar* dodijeliti dozvolu za stvaranje shemi na bazi podataka *studAdmin*. Kao korisnik *kolar* napraviti novu shemu *nastava*.

--postgres:

```
GRANT CREATE ON DATABASE studAdmin TO kolar;
```

--kolar

```
CREATE SCHEMA nastava;
```

Zadatak

d) Korisniku *kolar* omogućiti pregled, unos, izmjenu i brisanje svih podataka o predmetima (relacija *predmet*) s mogućnošću dodjeljivanja tih dozvola drugim korisnicima.

--postgres:

```
GRANT SELECT, INSERT, UPDATE, DELETE __ predmet ____ novak  
WITH _____;
```

Zadatak

d) Korisniku *kolar* omogućiti pregled, unos, izmjenu i brisanje svih podataka o predmetima (relacija *predmet*) s mogućnošću dodjeljivanja tih dozvola drugim korisnicima.

--postgres:

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

Zadatak 2

- baza podataka StudAdmin
- Spriječiti izmjenu studentove ocjene, ako je od njenog unosa prošlo više od 30 dana. Pokušaj izmjene spriječiti iznimkom koja signalizira pogrešku s tekstom 'Izmjena nije dozvoljena'.

Nadopunite

```
CREATE FUNCTION dojav iPogresku()  
  RETURNS TRIGGER AS ____  
BEGIN  
  RAISE _____ 'Izmjena ocjene više nije dozvoljena';  
  RETURN ____;  
END;  
$$ _____;
```

```
CREATE TRIGGER ocjenaUpd  
  BEFORE UPDATE ____ ocjena ____ upisanPredmet  
  FOR EACH ____  
  _____ ( OLD.datumOcjena + '30 days'::_____ < CURRENT_DATE)  
  EXECUTE PROCEDURE _____();
```

Rješenje

```
CREATE FUNCTION dojavipogresku()
```

```
  RETURNS TRIGGER AS $$
```

```
BEGIN
```

```
  RAISE EXCEPTION 'Izmjena ocjene više nije dozvoljena';
```

```
  RETURN NULL;
```

<- NULL: **otkazuju** se daljnje akcije

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER ocjenaUpd
```

```
  BEFORE UPDATE OF ocjena ON upisanPredmet
```

```
  FOR EACH ROW
```

```
  WHEN ( OLD.datumOcjena + '30 days'::INTERVAL < CURRENT_DATE)
```

```
    EXECUTE PROCEDURE dojavipogresku();
```