# SIGURNOST BAZE PODATAKA

XI predavanje

Dr.sc. Emir Mešković

# Integritet i sigurnost baze podataka

- Često se spominju zajedno, ali se radi o dva različita aspekta zaštite podataka
- Integritet baze podataka (*database integrity*)
  - Osigurava da su akcije koje korisnici pokušavaju izvesti ispravne (tj. uvijek rezultiraju konzistentnim stanjem baze podataka)
  - "podaci se štite od ovlaštenih korisnika"
- Sigurnost baze podataka (*database security*)
  - Osigurava da su korisnici ovlašteni za akcije koje pokušavaju izvesti
  - "podaci se štite od neovlaštenih korisnika"
- U oba slučaja:
  - moraju biti definisana pravila koja korisnici ne smiju narušiti
  - pravila se pohranjuju u rječnik podataka
  - SUBP nadgleda rad korisnika - osigurava poštivanje pravila

# Narušavanje sigurnosti i posljedice

- Oblici narušavanja sigurnosti baze podataka su:
  - Neovlašteno čitanje podataka
  - Neovlaštena izmjena podataka
  - Neovlašteno uništavanje podataka
- Moguće posljedice su:
  - Krađa ili prevara
  - Gubitak tajnosti
    - odnosi se na podatke kritične za funkcioniranje organizacije
    - npr. krađa recepture – rezultira gubitkom konkurentnosti na tržištu
  - Gubitak privatnosti
    - odnosi se na lične podatke
    - npr. krađa podataka o zdravstvenom stanju osobe – rezultira sudskim procesom proziv vlasnika baze podataka
  - Gubitak raspoloživosti
    - npr. uništenjem dijela podataka

# Protivmjere

- Sigurnost baze podataka se osigurava zaštitom na nekoliko nivoa
- Zaštita na nivou SUBP
  - Spriječiti pristup bazama podataka ili onim dijelovima baza podataka za koje korisnici nisu ovlašteni
- Zaštita na nivou operativnog sistema
  - Spriječiti pristup radnoj memoriji računara ili datotekama u kojima SUBP pohranjuje podatke
- Zaštita na nivou računarske mreže
  - Spriječiti presretanje poruka (sniffing) na internetu i intranetu
- Fizička zaštita
  - Fizički zaštititi lokaciju računarskog sistema
- Zaštita na nivou korisnika
  - Spriječiti da ovlašteni korisnici nepažnjom ili namjerno omoguće pristup podacima neovlaštenim osobama

# Aspekti zaštite podataka

- zakonski, socijalni i etički aspekt
  - da li vlasnik baze podataka ima zakonsko pravo na prikupljanje i korištenje podataka
- strategijski aspekt
  - tko definiše pravila pristupa – tko određuje kakve ovlasti ima pojedini korisnik baze podataka
- operativni aspekt
  - kako osigurati poštivanje pravila – kojim mehanizmima se osigurava poštivanje definisanih pravila, kako su zaštićene šifre (*password*), koliko često se mijenjaju

# Zakonska regulativa u BiH

- Zakon o zaštiti ličnih podataka („Sl. glasnik BiH“ broj: 49/06)
  - Cilj Zakona je da se na teritoriji Bosne i Hercegovine svim licima, bez obzira na njihovo državljanstvo ili prebivalište, osigura pravo na privatnost i zaštita njihovih ličnih podataka u postupku prikupljanja, obrade i korištenja ovih podataka.
  - Agencija za zaštitu ličnih podataka u Bosni i Hercegovini je samostalna upravna organizacija čija nadležnost i djelokrug poslova su propisani Zakonom o zaštiti ličnih podataka („Službeni glasnik BiH“ broj: 49/06).
- Evropska konvencija o zaštiti ljudskih prava i osnovnih sloboda („Sl. glasnik BiH“ broj: 6/99)

# Korisnici SUBP i provjera autentičnosti

- Administrator sistema (operativnog sistema ili SUBP) omogućuje korisniku pristup sistemu (operativnom sistemu ili SUBP) definiranjem jedinstvenog identifikatora korisnika (*user name*, *user ID*, *login ID*) i pripadajuće šifre (*password*) koja je poznata samo dotičnom korisniku i sitemu
- Korisnik koji pristupa sistemu (operativnom sistemu ili SUBP) poznavanjem šifre ovjerava svoju autentičnost (*authentication*)
- Za ovjeru autentičnosti korisnika SUBP može koristiti
  - mehanizme operativnog sistema ili
  - vlastite mehanizme

# Autorizacija i modeli kontrole pristupa

- Autorizacija je postupak kojim se određenom korisniku dodjeljuje dozvola za obavljanje određenih vrsta operacija (čitanje, izmjena, brisanje, ...) nad određenim objektima baze podataka (relacija, pogled, atribut, ...)
  - Podaci o dodijeljenim dozvolama pohranjuju se u riječnik podataka
- Prije obavljanja svake operacije, SUBP provjerava ima li korisnik dozvolu za obavljanje operacije nad objektom
  - Kontrola pristupa (*access control*)
- Današnji SUBP podržavaju dva različita modela kontrole pristupa podacima
  - Mandatna kontrola pristupa (MAC – *Mandatory Access Control*)
  - Diskrecijska kontrola pristupa (DAC – *Discretionary Access Control*)

# Mandatna kontrola pristupa

- Manji broj SUBP podržava mandatnu kontrolu pristupa
  - Koristi se relativno rijetko u odnosu na diskrecijsku kontrolu pristupa
- Mandatna kontrola pristupa je primjenjiva u sistemima u kojima se dozvole dodjeljuju na osnovu pozicije korisnika u hijerarhiji neke organizacije (vojska, državna uprava, ...)
- svaki objekat ima oznaku klasifikacijskog nivoa (*classification level*) – povjerljivo, tajno, vrlo tajno, ...
- svakom korisniku dodijeljena je oznaka nivoa ovlasti (*clearance level*)
  - neki objekat je dostupan korisnicima koji imaju odgovarajući nivo ovlasti

# Diskrecijska kontrola pristupa

- Većina današnjih SUBP podržava diskrecijsku kontrolu pristupa
  - Diskrecijska kontrola pristupa je podržana SQL standardom
- Korisnik ima različita prava pristupa (privilege, authority) različitim objektima
- Različiti korisnici imaju različita prava nad istim objektima
- Određenom korisniku se eksplicitno dodjeljuje dozvola za obavljanje određene operacije nad određenim objektom
  - Dozvole su opisane trojkama <korisnik, objekat, vrsta operacije>
  - Kada korisnik pokuša obaviti određenu operaciju nad određenim objektom (npr. Korisnik haso obavlja čitanje relacije stud) SUBP provjerava postoji li dozvola u obliku trojke <haso, stud, čitanje>

# Modeli kontrole pristupa

# Dozvole pristupa

- Korisnik dozvole
  - korisnik s određenom identifikacijskom oznakom (*userID*)
    - Pri uspostavljanju SQL sesije korisnik se prijavljuje svojim identifikatorom korisnika, te šifrom ovjerava svoju autentičnost
  - bilo koji korisnik (PUBLIC)
    - Dodjelom dozvole "korisniku" PUBLIC, dozvolu za obavljanje operacije dobivaju svi sadašnji i budući korisnici
- Objekti
  - baza podataka
  - tablica (relacija)
  - kolona tablice (atribut)
  - izvedena tablica (pogled, virtuelna tablica)
  - pohranjena procedura
- Vlasnik objekta
  - Vlasnik objekta je korisnik koji je kreirao objekat
  - Vlasnik objekta implicitno dobiva dozvole za obavljanje svih vrsta operacija nad objektom koje uključuju dodjeljivanje tih dozvola drugim korisnicima i uništavanje objekta

# Vrste dozvola u SQL-u

Za bazu podataka

- CONNECT
- CREATE
- DROP
- LOCK TABLES
- EVENT
- ALL (DBA)
- GRANT OPTION

Za tablicu

- SELECT [(cols)]
- INSERT [(cols)]
- DELETE
- UPDATE [(cols)]
- ALTER
- INDEX
- REFERENCES [(cols)]
- ALL
- CREATE TEMPORARY TABLES
- TRIGGER
- CREATE
- DROP
- GRANT OPTION

Za proceduru

- EXECUTE
- ALTER ROUTINE
- CREATE ROUTINE
- GRANT OPTION

- CREATE – kreiranje novih baza podataka i tabela
- DROP – brisanje postojećih baza podataka, tabela i pogleda
- LOCK TABLES – omogućava zaključavanje tabela sa izrazom LOCK TABLES (onih za koje ima SELECT dozvola)

- SELECT, INSERT, UPDATE - operacije nad svim ili samo određenim atributima
- DELETE - brisanje n-torki
- ALTER - izmjena strukture tablice, ograničenja
- INDEX - kreiranje i ukidanje indeksa
- REFERENCES - korištenje atributa pri definiciji stranih ključeva
- ALL – sve privilegije na datom nivou (osim GRANT OPTION)

- Za obavljanje naredbe:

```
INSERT INTO mjesto (Naziv)
   SELECT DISTINCT NazivMjesto
      FROM student
      WHERE NazivMjesto NOT IN
            (SELECT Naziv FROM mjesto);
```

- unos atributa Naziv u relaciji mjesto
- pregled atributa NazivMjesto u relaciji student
- pregled atributa Naziv u relaciji mjesto

# Matrica autorizacijskih pravila

| KORISNICI | OBJEKTI | | |
|---|---|---|---|
| | Relacija R | Atributi S.a S.b | Atributi S.c S.d S.e |
| PIRIC | SELECT<br>UPDATE<br>DELETE<br>where R.a >= 200 | SELECT<br><br>where S.a < 500 | NONE |
| DJURIC | NONE | NONE | NONE |
| PEJIC | SELECT<br>UPDATE<br>INSERT | SELECT | SELECT<br>UPDATE |

# Dodjeljivanje dozvola - MySQL

```
GRANT
   priv_type [(column list)][, priv_type [(column list)]]...
   ON [object_type] priv_level
   TO user_specification [, user_specification] ...
   [REQUIRE {NONE | tsl_option [[AND] tsl_option ...}]
   [WITH with_option ...]

object_type: TABLE | FUNCTION | PROCEDURE

priv_level: * | *.* | db:name.* | db_name.tbl_name | tbl_name |
            db_name.routine_name

user_specification: user [IDENTIFIED BY [PASSWORD] ‘password’

tsl_option: SSL | X509 | CHIPER ‘chiper’ | ISSUER ‘issuer’ |
             SUBJECT ‘subject’
with_option: GRANT OPTION | MAX_QUERIES_PER_HOUR count |
               MAX_UPDATES_PER_HOUR count |
               MAX_CONNECTIONS_PER_HOUR count |
               MAX_USER_CONNECTION count
```

# Ukidanje dozvola – MySQL

```
REVOKE
   priv_type [(column list)][, priv_type [(column list)]]...
   ON [object_type] priv_level
   FROM user [, user] ...

REVOKE ALL PRIVILAGES, GRANT OPTION
    FROM user [, user] ...
```

| KORISNICI | OBJEKTI | | |
|---|---|---|---|
| | Relacija R | Atributi S.a S.b | Atributi S.c S.d S.e |
| PIRIC | SELECT<br>UPDATE<br>DELETE<br>where R.a >= 200 | SELECT<br><br>where S.a < 500 | NONE |
| DJURIC | NONE | NONE | NONE |
| PEJIC | SELECT<br>UPDATE<br>INSERT | SELECT | SELECT<br>UPDATE |

```
CREATE VIEW R1 AS
    SELECT * FROM R
    WHERE R.a >= 200
     WITH CHECK OPTION;

CREATE VIEW S1 AS
     SELECT S.a, S.b FROM S
     WHERE S.a < 500
     WITH CHECK OPTION;
```

```
GRANT SELECT ON db.S1 TO ‘piric’@’%’;
GRANT SELECT, UPDATE, DELETE
      ON db.R1 TO ‘piric’@’%’;
GRANT SELECT, UPDATE, INSERT
      ON db.R TO ‘pejic’@’%’;
GRANT SELECT(a, b) ON db.S
      TO ‘pejic’@’%’;
GRANT SELECT(c,d,e),UPDATE(c, d, e)
      ON db.S TO ‘pejic’@’%’;
```

# Dodjeljivanje prenosivih dozvola

- Navođenje opcije WITH GRANT OPTION omogućuje korisniku koji je dobio dozvolu da je prenosi na druge korisnike
- Primjer:
  - Ako se korisniku “pejic" dodijeli dozvola:

```
GRANT SELECT, UPDATE, INSERT ON db.R TO ‘pejic’@’%’
WITH GRANT OPTION;
```

  - korisnik “pejic" može, nakon što je dobio prenosivu dozvolu, izvesti naredbu

```
GRANT SELECT ON db.R TO ‘djuric’@’%’;
```

# Ukidanje dozvola

- Navođenjem opcije CASCADE ukidaju se sve dozvole koje su dodijeljene samo na osnovu dozvole koja se ukida
- Primjer:
  - Ako je korisnik "pejic" na osnovi dozvole:

    ```
    GRANT SELECT, UPDATE, INSERT ON R TO pejic WITH
    GRANT OPTION;
    ```

  - dodijelio dozvolu

    ```
    GRANT SELECT ON R TO djuric;
    ```

  - Naredbom:

    ```
    REVOKE SELECT ON R FROM pejic CASCADE;
    ```

  - ukinut će se i korisniku "djuric" dozvola za pregled relacije R

# Ukidanje dozvola

- Navođenjem opcije RESTRICT ukidanje dozvola je onemogućeno ako su iz nje proizašle neke druge dozvole
- Primjer:
  - Ako je korisnik "pejic" na osnovi dozvole:

    ```
    GRANT SELECT, UPDATE, INSERT ON R TO pejic WITH
    GRANT OPTION;
    ```

  - dodijelio dozvolu

    ```
    GRANT SELECT ON R TO djuric;
    ```

  - Naredbom:

    ```
    REVOKE SELECT ON R FROM pejic RESTRICT;
    ```

  - Neće se ukinuti dozvola za pregled relacije R korisniku "pejic" niti korisniku "djuric"

Vlasnik tablica student i mjesto:

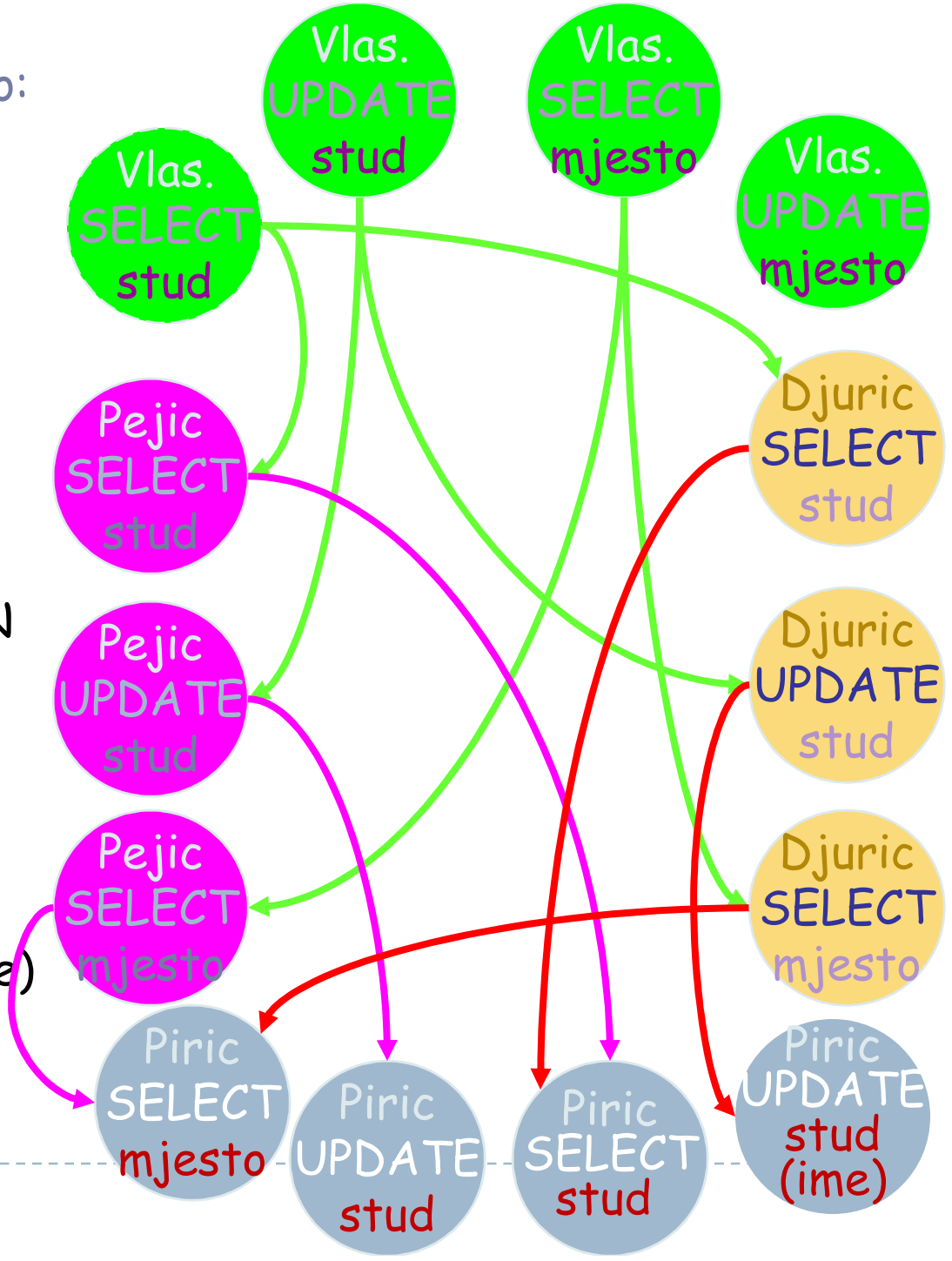
```
GRANT SELECT, UPDATE ON
   stud TO pejic, djuric
   WITH GRANT OPTION;

GRANT SELECT ON  mjesto
    TO pejic, djuric
    WITH GRANT OPTION;
```

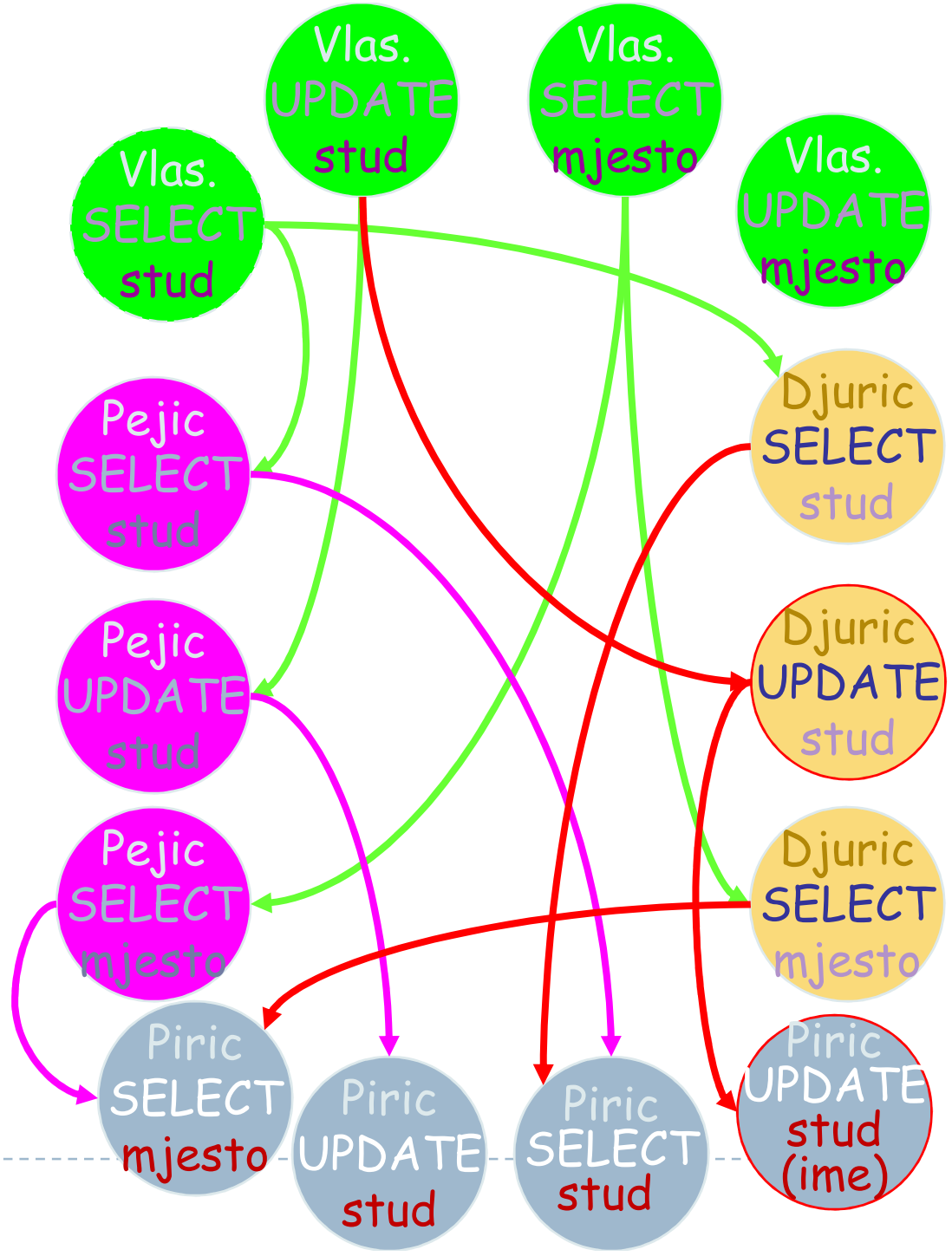
Pejic:

```
GRANT SELECT, UPDATE ON
    stud TO piric;
GRANT SELECT ON  mjesto
    TO piric;
```

Djuric:

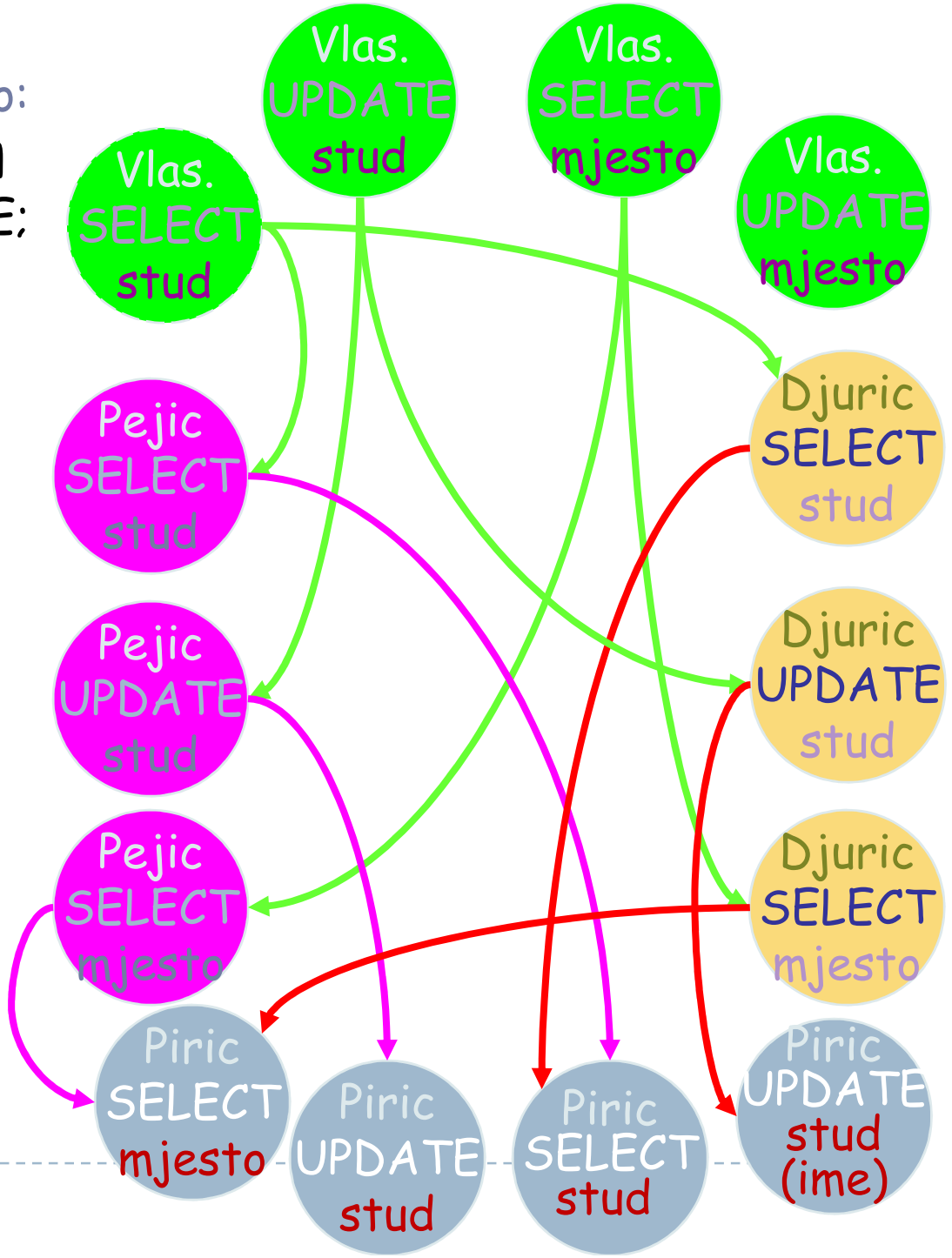
```
GRANT SELECT, UPDATE(ime)
    ON  stud TO piric;
GRANT SELECT ON  mjesto
    TO piric;
```




Vlas. UPDATE stud
Vlas. SELECT mjesto
Vlas. SELECT stud
Vlas. UPDATE mjesto
Djuric SELECT stud
Pejic SELECT stud
Djuric UPDATE stud
Pejic UPDATE stud
Djuric SELECT mjesto
Pejic SELECT mjesto
Piric SELECT mjesto
Piric UPDATE stud
Piric SELECT stud
Piric UPDATE stud (ime)

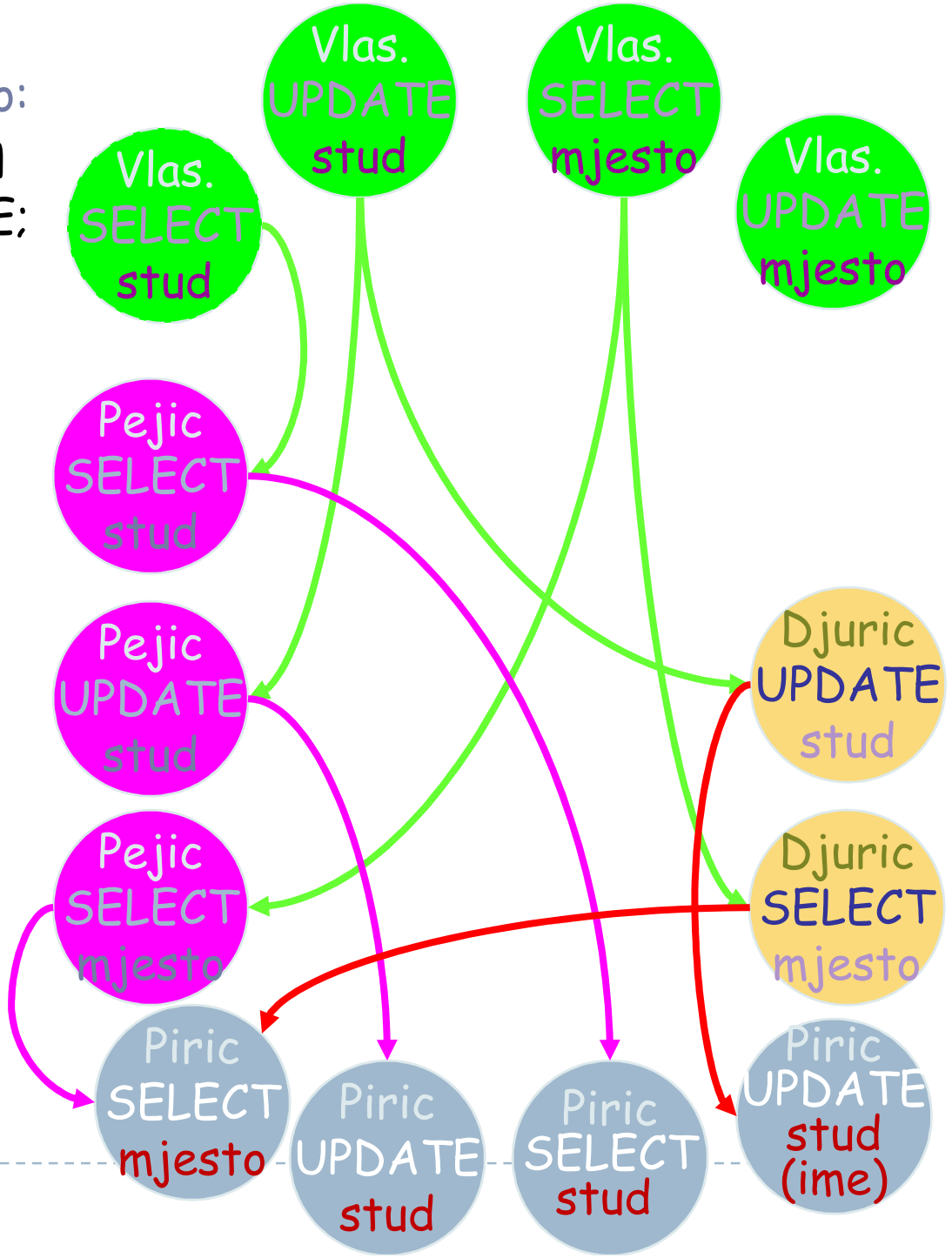
Vlasnik tablica student i mjesto:

REVOKE SELECT, UPDATE ON stud FROM djuric CASCADE;



Vlasnik tablica student i mjesto:

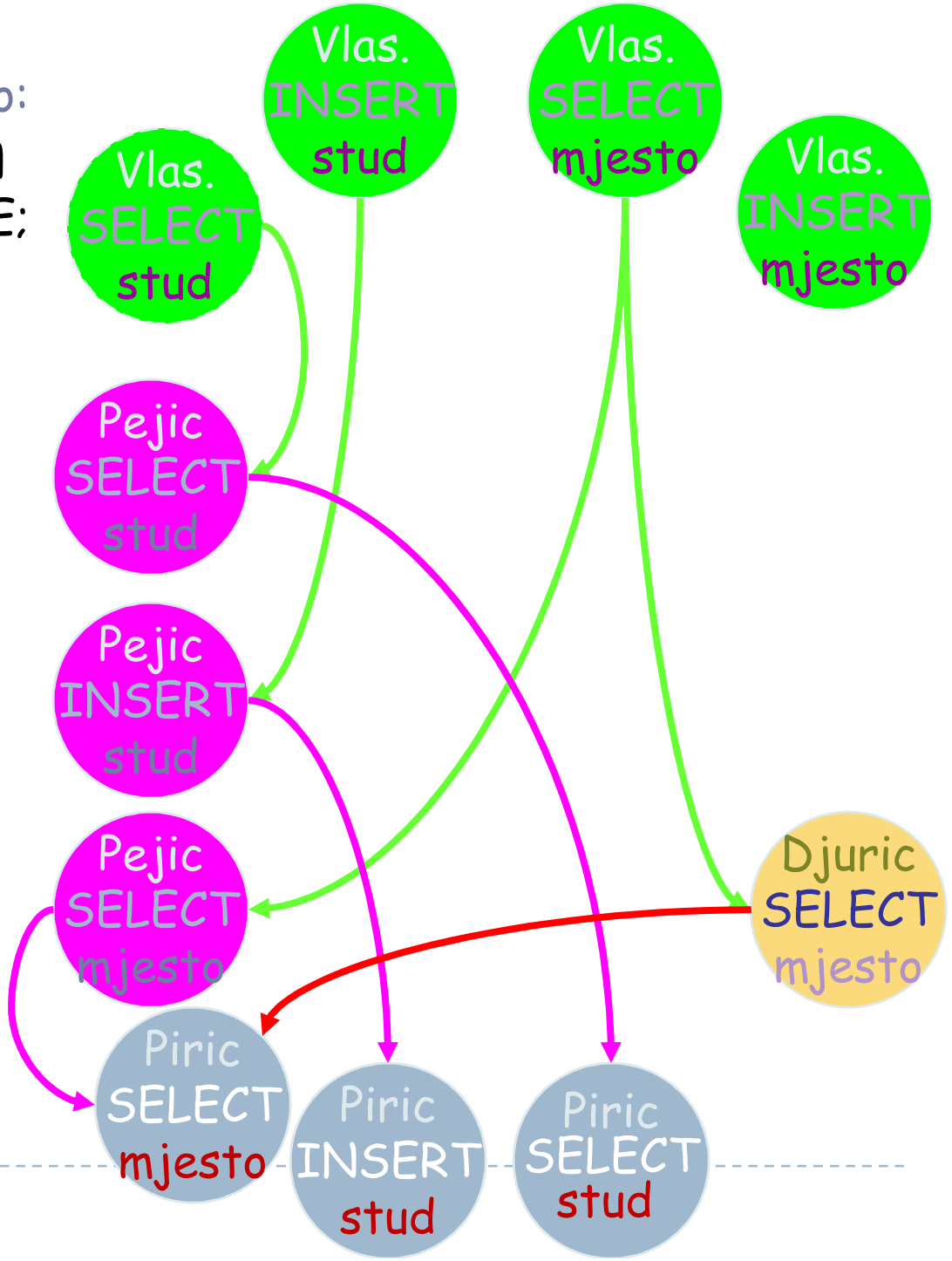
REVOKE SELECT, UPDATE ON stud FROM djuric CASCADE;



Vlasnik tablica student i mjesto:

REVOKE SELECT, UPDATE ON
stud FROM djuric CASCADE;

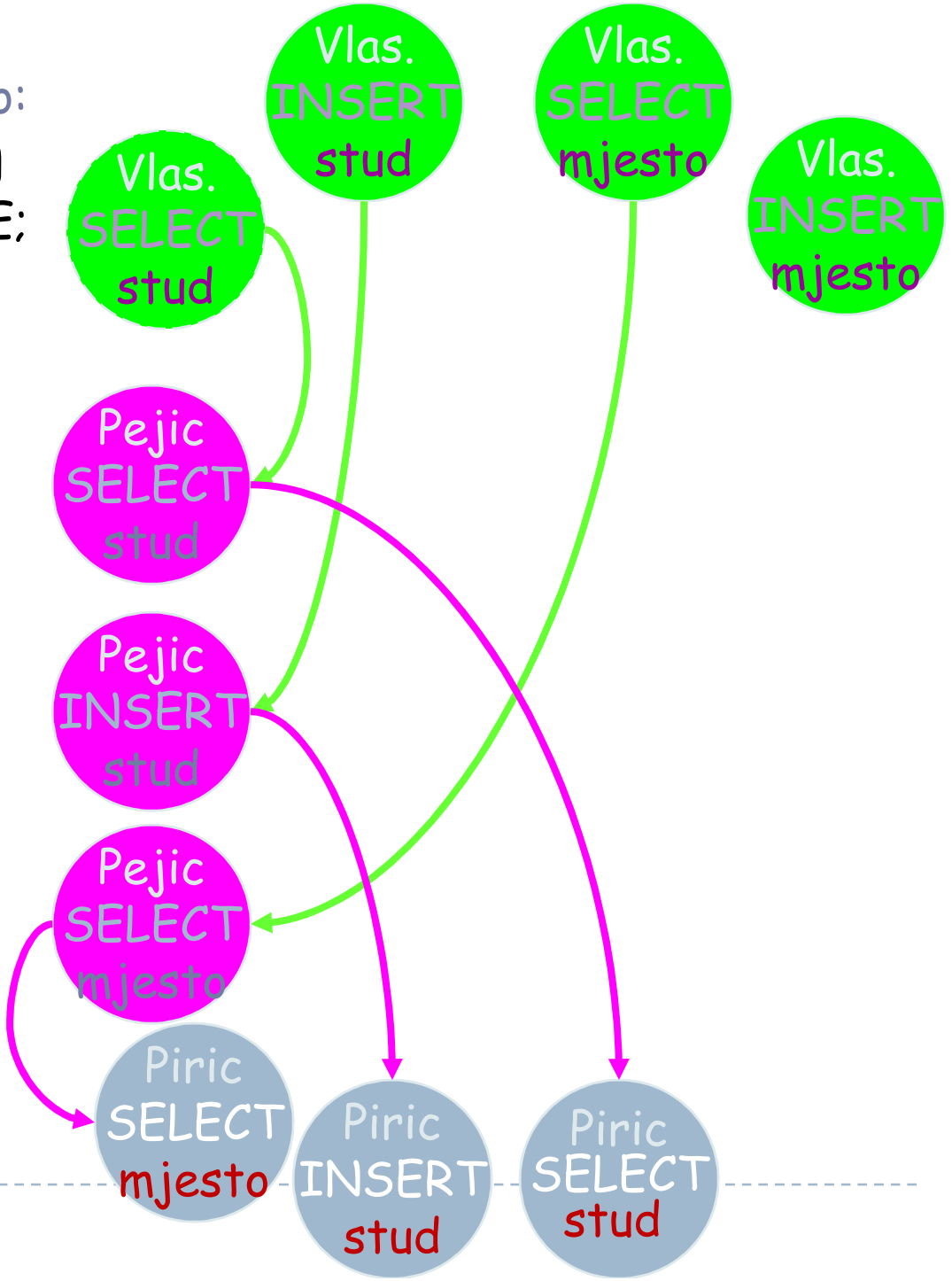
REVOKE SELECT ON mjesto
FROM djuric CASCADE;



Vlasnik tablica student i mjesto:

REVOKE SELECT, UPDATE ON
stud FROM djuric CASCADE;

REVOKE SELECT ON mjesto
FROM djuric CASCADE;



# Dodjeljivanje kontekstno ovisnih dozvola

- Primjer – dozvole nastavnicima nad tablicom ispit
  - Svaki nastavnik može pristupati samo svojim ispitima

    ISPIT = matBrSt, sifPred, datIspit, ocjena, sifNast
  - Možemo za svakog nastavnika definisati pogled i dati mu dozvole na pogled:

```
CREATE VIEW ispitNastPP01 AS
    SELECT * FROM ispit
      WHERE sifNast = 'PP01'
WITH CHECK OPTION;
REVOKE ALL ON db.ispit FROM ‘haso’@’%’;
GRANT ALL ON db.ispitNastPP01 TO ‘haso’@’%’;
```

  - Problemi:
    - broj pogleda (koliko ima nastavnika na FEu)
    - za svakog novog nastavnika treba kreirati novi pogled

# Dodjeljivanje kontekstno ovisnih dozvola

ISPIT = matBrSt, sifPred, datIspit, ocjena, sifNast

NAST = sifNast, imeNast, prezNast, userIdNast

- Kreiramo općeniti pogled:

```
CREATE VIEW ispitNast AS
   SELECT * FROM ispit
     WHERE sifNast IN
          (SELECT sifNast FROM nast
             WHERE userIdNast = CURRENT_USER)
WITH CHECK OPTION;
REVOKE ALL ON db.ispit FROM 'haso'@'%', 'huso'@'%';
GRANT ALL ON db.ispitNast TO 'haso'@'%';
GRANT ALL ON db.ispitNast TO 'huso'@'%';
```

Pitanje: Kakve dozvole na tablicu nast moze imati nastavnik??
npr. - smije li nastavnik mijenjati userIdNast???

# Dodjeljivanje istih dozvola velikom broju korisnika

- Primjer – dozvole bankovnim službenicama nad tablicom promet

  PROMET = brRacun, datVrijPromet, vrstaPromet, iznosPromet

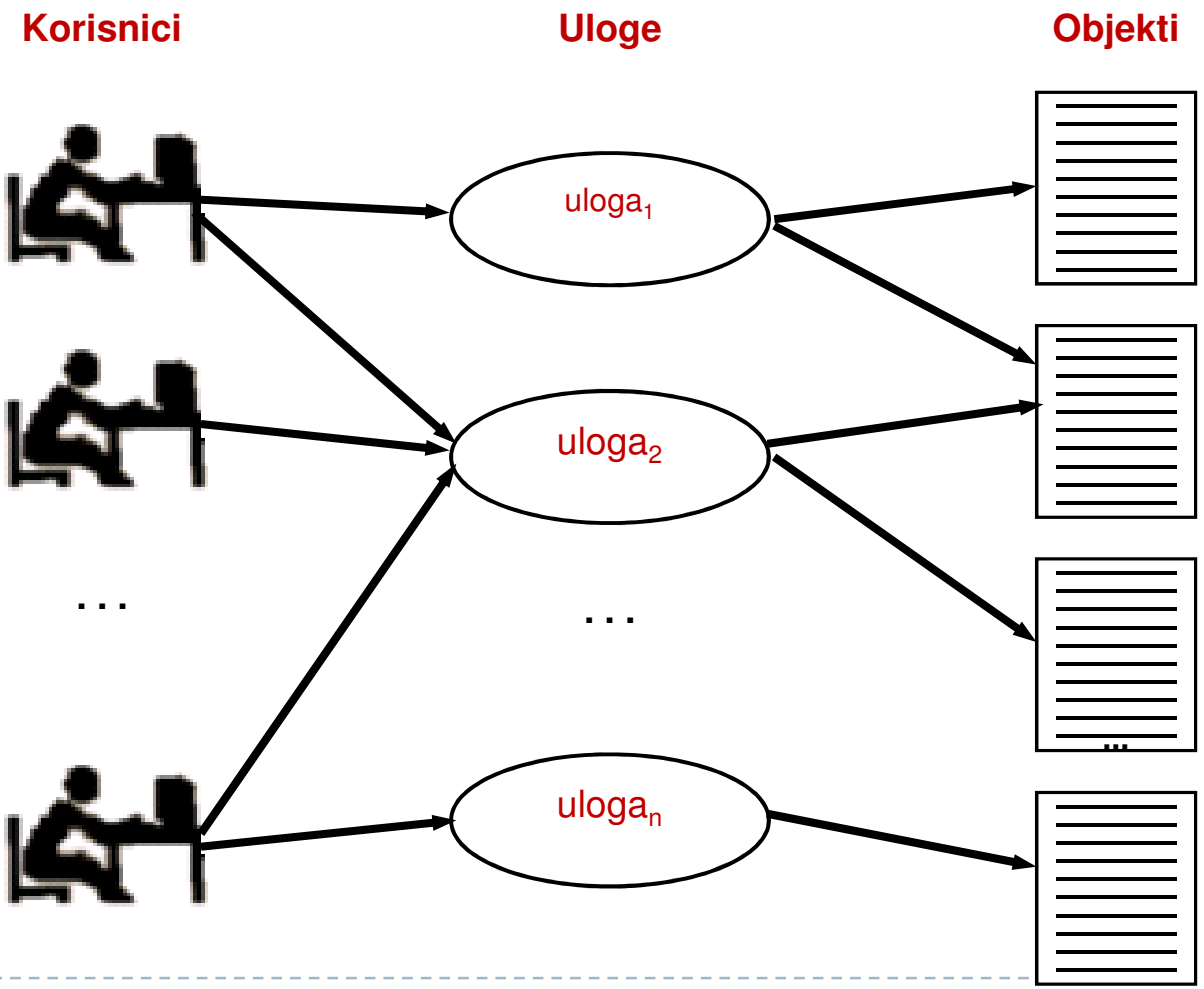
  - Svaki korisnik bi trebao dobiti dozvolu:

```
GRANT SELECT, INSERT, UPDATE ON promet TO haso;
GRANT SELECT, INSERT ON racun TO haso
GRANT SELECT, INSERT, UPDATE ON promet TO huso;
GRANT SELECT, INSERT ON racun TO huso;
GRANT SELECT, INSERT, UPDATE ON promet TO fata;
GRANT SELECT, INSERT ON racun TO fata;
```

  - Definišu se uloge (role)
  - Dozvole se dodjeljuju ulogama
  - Korisnicima se dodjeljuju uloge

```
CREATE ROLE sluzbenik;
GRANT SELECT, INSERT, UPDATE ON promet TO sluzbenik;
 GRANT SELECT, INSERT ON racun TO sluzbenik;
GRANT sluzbenik TO haso, huso, fata;
```

# Kontrola pristupa zasnovana na ulogama (*RBAC - Role Based Access Control*)



# Kontrola pristupa zasnovana na ulogama

- Jedan korisnik može imati više različitih uloga
- U jednom trenutku može biti aktivna samo jedna uloga korisnika
- Na početku rada ili u trenutku kad korisnik želi promijeniti ulogu s kojom želi raditi:

  SET ROLE roleName;

  - korisnik ima dozvole koje dobija preko uloge, vlastite dozvole – koje je direktno dobio i sve dozvole koje vrijede za sve korisnike (PUBLIC)

  SET ROLE NULL; ili

  SET ROLE NONE;

  - kada korisnik ne želi koristiti niti jednu ulogu
- Ukidanje uloge

  DROP ROLE roleName;
- Oduzimanje uloge korisniku

  REVOKE roleName FROM userName;

# Dozvola za izvođenje pohranjene rutine

- Korisnik koji kreira rutinu postaje vlasnik rutine
- vlasnik rutine se može definirati DEFINER klauzulom i mora imati sve dozvole za objekte i operacije u rutini
- vlasnik rutine može dodijeliti dozvolu za izvođenje rutine
- kada je SQL SECURITY karakteristika rutine DEFINER, korisnici koji imaju dozvolu za izvođenje rutine mogu izvršiti rutinu
- korisnik koji izvodi rutinu ne mora imati dozvolu za objekte i operacije koje se obavljaju tokom obavljanja rutine
- tokom izvođenja rutine vlasnikove dozvole se "prenose" na korisnika koji izvodi proceduru

# Dozvola za izvođenje pohranjene rutine

- Primjer: Korisnik piric obavlja naredbe:

```
CREATE PROCEDURE proc1 ()
    SQL SECURITY DEFINER
    ……
    INSERT INTO stud ……
    ……
END//
GRANT EXECUTE PROCEDURE ON db.proc1 TO 'djuric'@'%';
```

- piric ima dozvolu za unos u relaciju stud
- djuric nema dozvolu za unos u relaciju stud
- djuric ima dozvolu za obavljanje procedure proc1
- → djuric može kroz proceduru proc1 unijeti n-torke u relaciju stud !

# Dozvola za izvođenje pohranjene rutine

- Ako se prilikom kreiranja procedure navede SQL SECURITY karakteristika rutine INVOKER
  - bilo koji korisnik može izvesti proceduru
  - korisnik koji izvodi proceduru mora imati odgovarajuće dozvole za objekte i operacije koje se obavljaju unutar procedure

# Dozvola za izvođenje pohranjene rutine

- Primjer: Korisnik piric obavlja naredbe:

```
CREATE PROCEDURE proc1 ()
    SQL SECURITY INVOKER
    ……
    INSERT INTO stud ……
    ……
END//
GRANT EXECUTE PROCEDURE ON db.proc1 TO 'pejic'@'%', 'masic'@'%';
```

- pejic ima dozvolu za unos u relaciju stud
- masic nema dozvolu za unos u relaciju stud

→ pejic može kroz proc1 unijeti n-torke u relaciju stud !

→ masic ne može kroz proc1 unijeti n-torke u relaciju stud !

# Praćenje rada korisnika (*Auditing*)

- za osjetljive podatke može se evidentirati svaki pristup u posebnoj datoteci za praćenje rada korisnika *(Audit Trail)*
- tipičan zapis sadrži sljedeće informacije:
  - zahjev (naredba koja se izvršava , *statement source*)
  - mjesto s kojeg je upućen zahtjev (terminal, IP adresa računara)
  - korisnik (*user ID*) koji je pokrenuo operaciju
  - datum i vrijeme operacije
  - n-torke, atributi na koje se zahtjev odnosi
  - stara vrijednost
  - nova vrijednost
- sama činjenicom da se vodi "trag" često je dovoljna za sprečavanje zloupotrebe