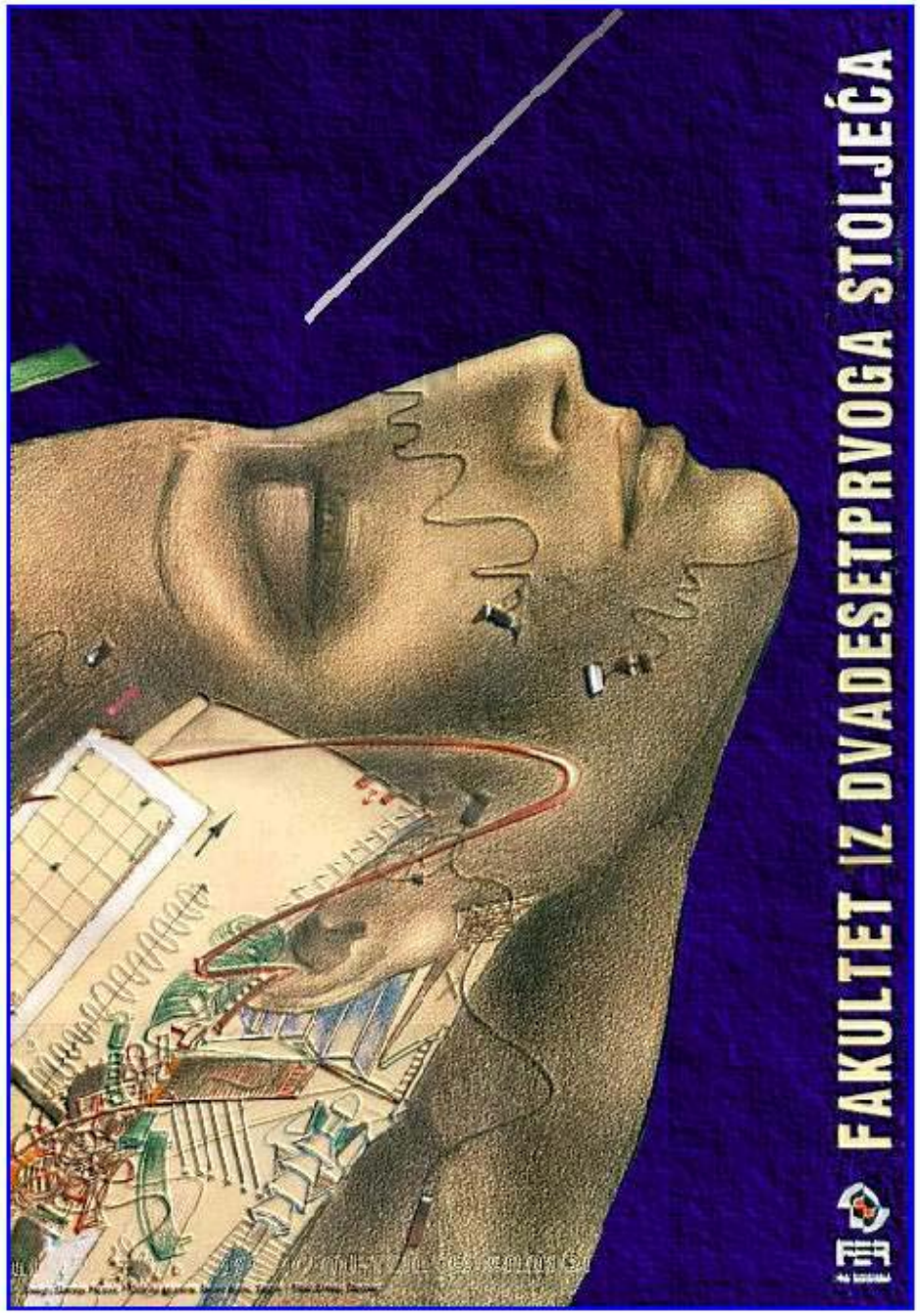
# Baze podataka

Predavanja
lipanj 2008.

# 15. Sigurnost baze podataka



# Integritet i sigurnost baze podataka

- Pojmovi integritet i sigurnost baze podataka se često spominju zajedno, međutim radi se o dva različita aspekta zaštite podataka
  - Integritet baze podataka (*database integrity*) - operacije nad podacima koje korisnici obavljaju **su ispravne** (tj. uvijek rezultiraju konzistentnim stanjem baze podataka)
    - "podaci se štite od ovlaštenih korisnika"
  - Sigurnost baze podataka (*database security*) - korisnici koji obavljaju operacije nad podacima **su ovlašteni** za obavljanje tih operacija
    - "podaci se štite od neovlaštenih korisnika"

Među ovim pojmovima postoje i sličnosti. U oba slučaja:

- moraju biti definirana **pravila** koja korisnici ne smiju narušiti
- pravila se pohranjuju u rječnik podataka
- SUBP nadgleda rad korisnika - osigurava poštivanje pravila

# Oblici narušavanja sigurnosti i moguće posljedice

- Oblici narušavanja sigurnosti baze podataka su:
  - neovlašteno čitanje podataka
  - neovlaštena izmjena podataka
  - neovlašteno uništavanje podataka
- Moguće posljedice su:
  - krađa ili prijevara
  - gubitak tajnosti
    - odnosi se na podatke kritične za funkcioniranje organizacije
    - npr. krađa recepture - rezultira gubitkom konkurentnosti na tržištu
  - gubitak privatnosti
    - odnosi se na osobne podatke
    - npr. krađa podataka o zdravstvenom stanju osobe - rezultira sudskim procesom protiv vlasnika baze podataka
  - gubitak raspoloživosti
    - npr. uništenjem dijela podataka

# Protumjere

- sigurnost baze podataka se osigurava zaštitom na nekoliko razina
  - **zaštita na razini SUBP**
    - spriječiti pristup bazama podataka ili onim dijelovima baza podataka za koje korisnici nisu ovlašteni
  - **zaštita na razini operacijskog sustava**
    - spriječiti pristup radnoj memoriji računala ili datotekama u kojima SUBP pohranjuje podatke
  - **zaštita na razini računalne mreže**
    - spriječiti presretanje poruka (*sniffing*) na internetu i intranetu
  - **fizička zaštita**
    - fizički zaštititi lokaciju računalnog sustava
  - **zaštita na razini korisnika**
    - spriječiti da ovlašteni korisnici nepažnjom ili namjerno (npr. u zamjenu za mito ili druge usluge) omoguće pristup podacima neovlaštenim osobama

# Aspekti zaštite podataka

- **zakonski, socijalni i etički aspekt**
  - ima li vlasnik baze podataka zakonsko pravo na prikupljanje i korištenje podataka
  - npr. smije li zdravstvena ustanova koja, u skladu sa zakonom prikuplja podatke o pacijentima, te iste podatke koristiti pri donošenju odluke hoće li svog bivšeg pacijenta zaposliti
- **strategijski aspekt**
  - tko definira pravila pristupa - tko određuje kakve ovlasti ima pojedini korisnik baze podataka, ...
- **operativni aspekt**
  - kako osigurati poštivanje pravila - kojim mehanizmima se osigurava poštivanje definiranih pravila, na koji način su lozinke zaštićene, koliko često se mijenjaju, ...

# Ustav RH - Članak 37.

Svakom se jamči sigurnost i tajnost osobnih podataka. Bez privole ispitanika, osobni se podaci mogu prikupljati, obrađivati i koristiti samo uz uvjete određene zakonom.

Zakonom se uređuje zaštita podataka te nadzor nad djelovanjem informatičkih sustava u Republici.

Zabranjena je uporaba osobnih podataka suprotna utvrđenoj svrsi njihovoga prikupljanja.

- Zakon o zaštiti osobnih podataka

# Korisnici SUBP i ovjera autentičnosti

- administrator sustava (operacijskog sustava ili SUBP) omogućuje korisniku pristup sustavu (operacijskom sustavu ili SUBP) definiranjem jedinstvenog identifikatora korisnika (*user name*, *user ID*, *login ID*) i pripadne lozinke (*password*) koja je poznata samo dotičnom korisniku i sustavu
- korisnik koji pristupa sustavu (operacijskom sustavu ili SUBP) poznavanjem lozinke ovjerava svoju autentičnost (*authentication*)
- za ovjeru autentičnosti korisnika SUBP može koristiti
  - mehanizme operacijskog sustava

    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 rječnik 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 sustavima u kojima se dozvole dodjeljuju na temelju pozicije korisnika u hijerarhiji neke organizacije (vojska, državna uprava, ...)

- svaki **objekt** dobiva oznaku klasifikacijske razine (*classification level*), npr. povjerljivo, tajno, vrlo tajno, ...
- svakom **korisniku** dodjeljuje se oznaka razine ovlasti (*clearance level*)
  - korisnici mogu obavljati operacije nad onim objektima za koje imaju odgovarajuću razinu ovlasti

# Diskrecijska kontrola pristupa

- većina današnjih SUBP podržava diskrecijsku kontrolu pristupa
  - diskrecijska kontrola pristupa je podržana SQL standardom
- određenom korisniku se eksplicitno dodjeljuje dozvola za obavljanje određene operacije nad određenim objektom
  - dozvole su opisane trojkama <korisnik, objekt, vrsta operacije>

    <horvat, ispit, čitanje>

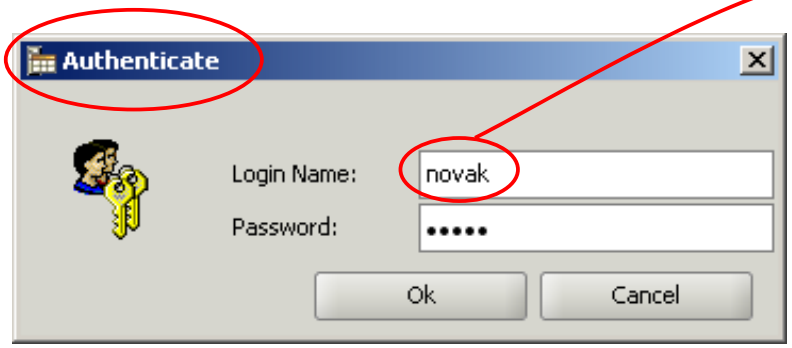
    <horvat, ispit, izmjena>

    <horvat, predmet, čitanje>

    <novak, predmet, čitanje>
  - kada korisnik novak pokuša obaviti operaciju čitanja objekta (relacije) predmet, SUBP provjerava postoji li dozvola u obliku trojke <novak, predmet, čitanje>
- u preostalom dijelu predavanja razmatrat će se diskrecijska kontrola pristupa

# Korisnici u SQL-u

- korisnik s određenom identifikacijskom oznakom (*userID*)
  - pri uspostavljanju SQL-sjednice korisnik se prijavljuje svojim identifikatorom korisnika, te lozinkom ovjerava svoju autentičnost
  - funkcija USER vraća vrijednost identifikatora korisnika koji se koristi u dotičnoj SQL-sjednici




```
SELECT FIRST 1 USER AS korisnik
  FROM mjesto;
```

| korisnik |
| --- |
| novak |

- bilo koji korisnik (PUBLIC)
  - dodjelom dozvole "korisniku" PUBLIC, dozvolu za obavljanje operacije dobivaju svi sadašnji i budući korisnici

# Objekti i vlasnici objekata u SQL-u

- **Objekti**
  - relacija (tablica, *table*)
  - atribut (stupac tablice, *column*)
  - virtualna relacija (pogled, *view*)
  - baza podataka

- **Vlasnik objekta (*object owner*)**
  - vlasnik objekta je korisnik koji je kreirao objekt, npr:
    - vlasnik baze podataka je korisnik koji je kreirao bazu podataka
    - vlasnik relacije je korisnik koji je kreirao relaciju
  - vlasnik objekta implicitno dobiva dozvole za obavljanje **svih** vrsta operacija nad objektom, uključujući dozvole za:
    - dodjeljivanje svih vrsta dozvola nad tim objektom drugim korisnicima
    - uništavanje objekta

# Vrste dozvola u SQL-u na razini baze podataka (*dbPrivilege*)

- Različiti SUBP imaju različita rješenja za dodjeljivanje dozvola na razini baze podataka. Ovdje je prikazano rješenje koje se koristi u sustavu IBM Informix:
  - **CONNECT**
    - uspostavljanje SQL-sjednice i obavljanje operacija nad objektima za koje je korisnik dobio dozvolu od vlasnika objekta ili je njihov vlasnik, kreiranje virtualnih i privremenih relacija
  - **RESOURCE**
    - CONNECT + kreiranje **novih** relacija u bazi podataka
  - **DBA**
    - RESOURCE + neovisno o vlasništvu i dozvolama nad objektima u bazi podataka: sve vrste operacija nad svim objektima, uništavanje svih objekata (uključujući i bazu podataka)
    - korisnik koji kreira bazu podataka je vlasnik te baze podataka i implicitno dobiva DBA (*Database administrator*) dozvolu

# Vrste dozvola u SQL-u na razini [virtualne] relacije (*tablePrivilege*)

- **SELECT [(*columnList*)]**
  - čitanje n-torki (ili vrijednosti navedenih atributa) [virtualne] relacije
- **UPDATE [(*columnList*)]**
  - izmjena n-torki (ili vrijednosti navedenih atributa) [virtualne] relacije
- **INSERT**
  - unos n-torki [virtualne] relacije
- **DELETE**
  - brisanje n-torki [virtualne] relacije
- **REFERENCES [(*columnList*)]**
  - korištenje **relacije** (ili samo navedenih atributa kao pozivane relacije pri definiranju stranog ključa)
- **INDEX**
  - kreiranje indeksa nad **relacijom**
- **ALTER**
  - izmjena strukture **relacije** i definiranje integritetskih ograničenja
- **ALL PRIVILEGES**
  - sve do sada navedene vrste operacija nad [virtualnom] relacijom

# SQL naredbe za dodjeljivanje i ukidanje dozvola

- GRANT *dbPrivilege* TO { PUBLIC | *userList* }
- REVOKE *dbPrivilege* FROM { PUBLIC | *userList* }
- GRANT *tablePrivilegeList* ON { *tableName* | *viewName* }
  TO { PUBLIC | *userList* | *roleList* }
  [ WITH GRANT OPTION ]
- REVOKE *tablePrivilegeList* ON { *tableName* | *viewName* }
  FROM { PUBLIC | *userList* | *roleList* }
  [ CASCADE | RESTRICT ]

# Primjer 1:

student

| matBr | ime | prez | pbr | adresa |
|---|---|---|---|---|
| 100 | Ana | Ivić | 51000 | Korzo 2 |
| 102 | Ivan | Perić | 10000 | Ilica 20 |
| 105 | Matija | Matić | 31000 | Unska 7 |
| 107 | Tea | Bilić | 10000 | Vlaška 5 |

ispit

| matBr | nazPred | datIsp | ocj |
|---|---|---|---|
| 100 | Fizika | 1.5.2004 | 3 |
| 102 | Matematika | 7.9.2003 | 1 |
| 102 | Matematika | 9.2.2004 | 5 |
| 107 | Fizika | 5.4.2006 | 4 |

- kreirati bazu podataka studBaza i relacije student i ispit
  - vlasnik baze podataka i relacija treba biti korisnik bpadmin
- korisnik horvat treba dobiti dozvole:
  - pregled svih podataka u relacijama student i ispit
  - unos, izmjena, brisanje svih podataka u relaciji ispit
- korisnik novak treba dobiti dozvole:
  - pregled svih podataka u relaciji student
  - izmjena poštanskog broja i adrese u relaciji student
- korisnik kolar treba dobiti dozvolu:
  - pregled svih podataka u relaciji student, osim adrese

# Primjer 1 (nastavak):

bpadmin ← naredbe obavlja korisnik bpadmin

```
CREATE DATABASE studBaza;
CREATE TABLE student (...);
CREATE TABLE ispit (...);

GRANT CONNECT TO horvat;
GRANT CONNECT TO novak;
GRANT CONNECT TO kolar;

GRANT SELECT ON student
  TO horvat;
GRANT SELECT, INSERT
    , UPDATE, DELETE ON ispit
  TO horvat;

GRANT SELECT ON student
  TO novak;
GRANT UPDATE(pbr, adresa)
  ON student TO novak;

GRANT SELECT(matBr, ime
           , prez, pbr)
  ON student TO kolar;
```

- korisnik bpadmin je vlasnik baze podataka studBaza i relacija student i ispit. Posjeduje DBA dozvolu na razini baze podataka
- dozvole za uspostavljanje SQL-sjednice
- dozvole korisniku horvat za pregled podataka u relaciji student
- dozvole korisniku horvat za pregled, unos, izmjenu i brisanje podataka u relaciji ispit
- dozvola korisniku novak za pregled podataka u relaciji student
- dozvola korisniku novak za izmjenu vrijednosti atributa u relaciji student
- dozvola korisniku kolar za pregled svih podataka u relaciji student, osim adrese

# Primjer 2:

**bpadmin**

```
CREATE DATABASE studBaza;
GRANT RESOURCE TO horvat;
GRANT CONNECT TO novak;
```

korisnik bpadmin kreira bazu podataka studBaza. Kao vlasnik baze podataka implicitno dobiva DBA dozvolu na razini baze podataka

**horvat**

```
CREATE TABLE zupanija (
  sifZup INTEGER
, nazZup CHAR(30)
, PRIMARY KEY(sifZup));
GRANT SELECT, INSERT, UPDATE
  ON zupanija TO novak;
```

može jer ima RESOURCE dozvolu

može jer je vlasnik relacije zupanija

**novak**

```
SELECT * FROM zupanija;
INSERT INTO zupanija ...;
UPDATE zupanija ...;
```

može jer ima barem CONNECT dozvolu (bez CONNECT dozvole ne bi mogao uspostaviti SQL-sjednicu), te dozvole koje je dobio od vlasnika relacije zupanija

# Primjer 2 (nastavak):

**novak**

```
DROP TABLE zupanija;
```

ne može jer nije vlasnik objekta niti ima DBA dozvolu

**kolar**

```
SELECT * FROM zupanija;
```

ne može jer nema niti CONNECT dozvolu (ne može uspostaviti SQL-sjednicu)

**horvat**

```
GRANT CONNECT TO kolar;
```

ne može jer nema DBA dozvolu

**bpadmin**

```
GRANT CONNECT TO kolar;
```

može jer ima DBA dozvolu

**horvat**

```
GRANT SELECT
  ON zupanija TO kolar;
```

može jer je vlasnik relacije zupanija

**kolar**

```
SELECT * FROM zupanija;
```

može jer ima barem CONNECT dozvolu, te dozvolu za obavljanje operacije SELECT nad relacijom zupanija

# Primjer 2 (nastavak):

**novak**

```
CREATE TABLE mjesto ...;
```

ne može jer nema RESOURCE dozvolu

**horvat**

```
GRANT RESOURCE TO novak;
```

ne može jer nema DBA dozvolu

**bpadmin**

```
GRANT DBA TO horvat;
```

može jer ima DBA dozvolu

**horvat**

```
GRANT RESOURCE TO novak;
```

može jer ima DBA dozvolu

**novak**

```
CREATE TABLE mjesto (...
  REFERENCES zupanija ...);
```

ne može jer nema dozvolu za kreiranje stranog ključa koji se poziva na primarni ključ relacije zupanija (mogao bi kreirati relaciju bez stranog ključa jer ima RESOURCE dozvolu)

# Primjer 2 (nastavak):

**horvat**

```
GRANT REFERENCES
  ON zupanija TO novak;
```

može jer ima DBA dozvolu (ali čak i da nema DBA dozvolu, vlasnik je relacije zupanija)

**novak**

```
CREATE TABLE mjesto ...(
  REFERENCES zupanija ...);
```

može jer ima RESOURCE dozvolu i dozvolu za kreiranje stranog ključa koji se poziva na primarni ključ relacije zupanija

**horvat**

```
GRANT CONNECT TO PUBLIC;
```

može jer ima DBA dozvolu

- sada svaki korisnik (sadašnji ili budući) koji uspije ovjeriti svoju autentičnost može uspostaviti SQL-sjednicu s bazom podataka studBaza

**novak**

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

može jer je vlasnik relacije mjesto

- sada svaki korisnik (sadašnji ili budući) koji uspostavi SQL-sjednicu s bazom podataka (uz prethodnu ovjeru autentičnosti) može obavljati operaciju SELECT nad relacijom mjesto

# Dodjeljivanje prenosivih dozvola

- ukoliko se korisniku dozvola dodijeli uz navođenje opcije WITH GRANT OPTION, korisnik će moći dodjeljivati tu istu dozvolu ostalim korisnicima (unatoč tome što nije vlasnik objekta)

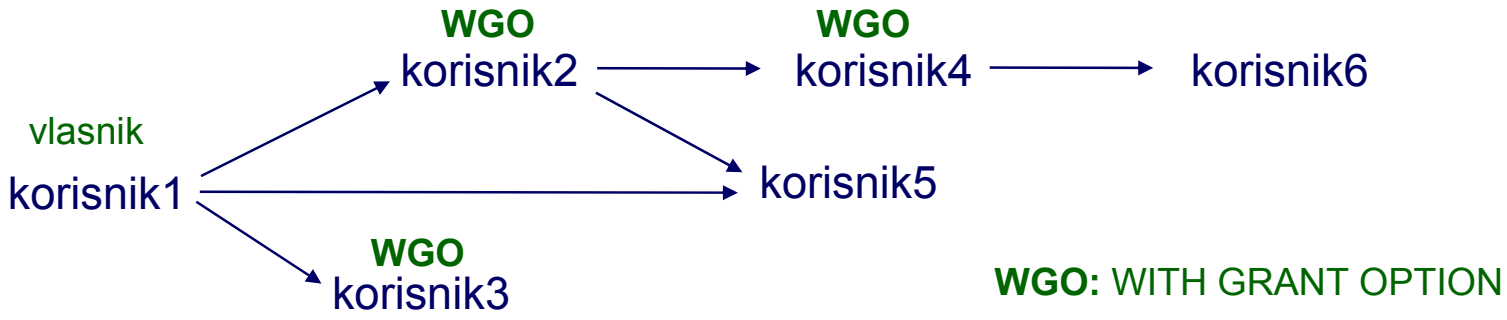
Primjer:

**korisnik1**
```
CREATE TABLE ispit (...);
GRANT SELECT ON ispit TO korisnik2 WITH GRANT OPTION;
GRANT SELECT ON ispit TO korisnik3 WITH GRANT OPTION;
```

**korisnik2**
```
GRANT SELECT ON ispit TO korisnik4 WITH GRANT OPTION;
GRANT SELECT ON ispit TO korisnik5;
```

**korisnik4**
```
GRANT SELECT ON ispit TO korisnik6;
```

**korisnik1**
```
GRANT SELECT ON ispit TO korisnik5;
```



# Ukidanje dozvola

- korisnik koji je dozvolu dodijelio, tu istu dozvolu može ukinuti naredbom REVOKE

Primjer:
- vlasnik baze podataka studBaza je korisnik bpadmin
- vlasnik relacije mjesto je korisnik horvat

**horvat**

```
GRANT SELECT, UPDATE ON mjesto TO novak WITH GRANT OPTION;
```

**novak**

```
GRANT SELECT, UPDATE ON mjesto TO kolar;
```

- npr. naredbu:

```
REVOKE UPDATE ON mjesto FROM kolar;
```

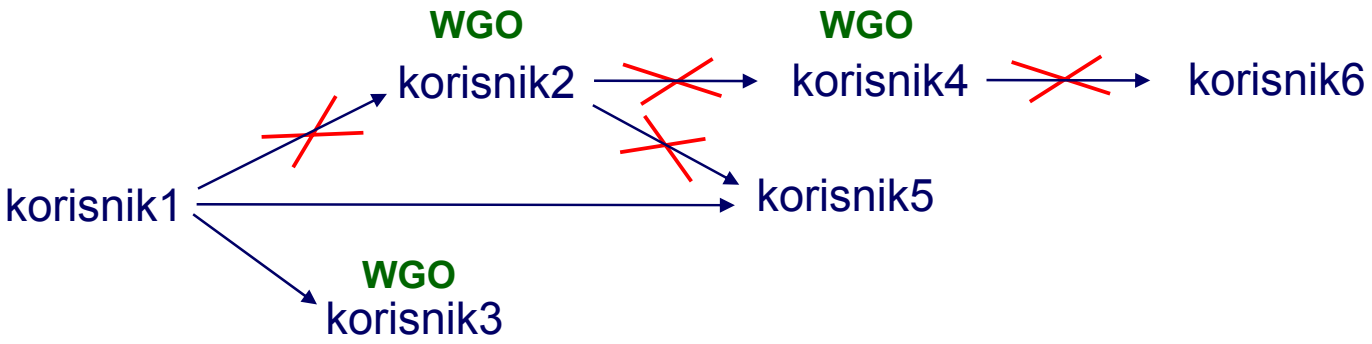
- može obaviti korisnik novak jer je novak korisnik koji je dozvolu dodijelio

# Ukidanje dozvola dodijeljenih temeljem WITH GRANT OPTION

- ukidanjem dozvole korisniku x (koji je dozvole dalje dodjeljivao temeljem ovlasti stečene pomoću WITH GRANT OPTION) uz primjenu opcije CASCADE, dozvola se ukida i svim ostalim korisnicima koji su dotičnu dozvolu stekli od korisnika x (neposredno ili posredno)

Primjer: `korisnik1`

```
REVOKE SELECT ON ispit FROM korisnik2 CASCADE;
```

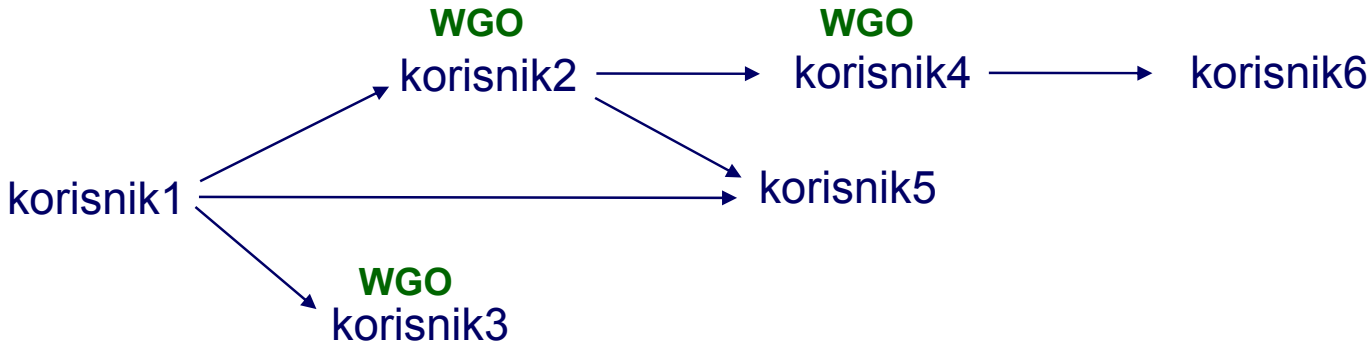



- obavljanjem naredbe dozvolu gube korisnik2, korisnik4 i korisnik6
- korisnik5 će izgubiti dozvolu koju je dobio od korisnika2, ali će zadržati dozvolu koju je dobio od korisnika1
- ukoliko se opcija CASCADE ne navede, naredba REVOKE djeluje na jednak način kao kada je opcija CASCADE navedena

# Ukidanje dozvola dodijeljenih temeljem WITH GRANT OPTION

- ukidanjem dozvole korisniku x uz primjenu opcije RESTRICT, dozvola će biti ukinuta jedino u slučaju kada korisnik x nije dalje dodjeljivao ovlasti temeljem ovlasti stečene pomoću WITH GRANT OPTION

Primjer:




**korisnik1**

```
REVOKE SELECT ON ispit FROM korisnik2 RESTRICT;
```

SUBP odbija obaviti naredbu (dojavljuje pogrešku)

**korisnik2**

```
REVOKE SELECT ON ispit FROM korisnik4 RESTRICT;
```

SUBP odbija obaviti naredbu (dojavljuje pogrešku)

**korisnik1**

```
REVOKE SELECT ON ispit FROM korisnik3 RESTRICT;
```

SUBP obavlja naredbu (korisnik3 ostaje bez dozvole)

# Primjena virtualnih relacija

ispit

| mbrSt | nazPred | datIsp | ocj |
|---|---|---|---|
| 100 | Fizika | 1.5.2004 | 3 |
| 102 | Matematika | 7.9.2003 | 1 |
| 102 | Matematika | 9.2.2004 | 5 |
| 107 | Fizika | 5.4.2006 | 4 |

- vlasnik relacije ispit je korisnik horvat
- korisniku novak omogućiti pregled samo prosječnih ocjena po predmetima
- korisniku kolar omogućiti pregled, unos, izmjenu i brisanje samo za ispite iz predmeta Fizika

**horvat**

```
CREATE VIEW prosjek (nazPred, prosOcj) AS
  SELECT nazPred, AVG(ocj)
    FROM ispit
    GROUP BY nazPred;
GRANT SELECT ON prosjek TO novak;

CREATE VIEW ispitFizika AS
  SELECT * FROM ispit
    WHERE nazPred = 'Fizika'
  WITH CHECK OPTION;
GRANT SELECT, INSERT, UPDATE, DELETE
  ON ispitFizika TO kolar;
```

- zašto je nužno virtualnu relaciju ispitFizika kreirati uz opciju WITH CHECK OPTION?!

# Dodjeljivanje kontekstno ovisnih dozvola

ispit

| mbrSt | sifPred | datIsp | ocj |
|---|---|---|---|
| 100 | 100 | 1.5.2004 | 3 |
| 102 | 200 | 7.9.2003 | 1 |
| 102 | 200 | 9.2.2004 | 5 |
| 107 | 300 | 5.4.2006 | 4 |

nast

| sifNast | imeN | prezN | userId |
|---|---|---|---|
| 1001 | Slavko | Kolar | kolar |
| 1002 | Ivo | Ban | ban |
| 1003 | Ana | Novak | novak |

predaje

| sifNast | sifPred |
|---|---|
| 1001 | 100 |
| 1001 | 200 |
| 1002 | 200 |
| 1003 | 200 |
| 1003 | 300 |

- vlasnik relacija je korisnik horvat
- svakom nastavniku (korisnicima kolar, ban, novak) omogućiti pregled i izmjenu ispita samo iz predmeta koje predaju

**LOŠE RJEŠENJE!**

horvat

```
CREATE VIEW kolarIspiti AS
  SELECT * FROM ispit
    WHERE sifPred IN (
      SELECT sifPred FROM predaje
        WHERE sifNast = 1001) WITH CHECK OPTION;
GRANT SELECT, UPDATE ON kolarIspiti TO kolar;
```

- ponoviti za svakog nastavnika: banIspiti, novakIspiti, ...
- nova virtualna relacija za svakog novog nastavnika (≈150 na FER-u)
- svaki nastavnik upit nad relacijom ispit mora pisati na drugačiji način

# Dodjeljivanje kontekstno ovisnih dozvola

ispit

| mbrSt | sifPred | datIsp | ocj |
|---|---|---|---|
| 100 | 100 | 1.5.2004 | 3 |
| 102 | 200 | 7.9.2003 | 1 |
| 102 | 200 | 9.2.2004 | 5 |
| 107 | 300 | 5.4.2006 | 4 |

nast

| sifNast | imeN | prezN | userId |
|---|---|---|---|
| 1001 | Slavko | Kolar | kolar |
| 1002 | Ivo | Ban | ban |
| 1003 | Ana | Novak | novak |

predaje

| sifNast | sifPred |
|---|---|
| 1001 | 100 |
| 1001 | 200 |
| 1002 | 200 |
| 1003 | 200 |
| 1003 | 300 |

**ISPRAVNO RJEŠENJE!**

horvat

```
CREATE VIEW ispitiZaNastavnike AS
  SELECT * FROM ispit
    WHERE sifPred IN (
      SELECT sifPred FROM predaje, nast
       WHERE predaje.sifNast = nast.sifNast
         AND userId = USER) WITH CHECK OPTION;
GRANT SELECT, UPDATE ON ispitiZaNastavnike TO kolar;
GRANT SELECT, UPDATE ON ispitiZaNastavnike TO ban;
GRANT SELECT, UPDATE ON ispitiZaNastavnike TO novak;
```

- "sadržaj" virtualne relacije ovisit će o identifikatoru nastavnika koji je ostvario SQL-sjednicu
- smije li se nastavnicima dozvoliti izmjena vrijednosti atributa userId u relaciji nast ili sadržaj relacije predaje?!

# Upotreba sinonima

PROBLEM:

- nastavnici (odnosno aplikativni ili primjenski programi koje nastavnici koriste) moraju u upitima o ispitima koristiti virtualnu relaciju ispitiZaNastavnike

```
SELECT * FROM ispitiZaNastavnike WHERE ocj = 1;
```

- dekan (npr. korisnik s identifikatorom novosel), za razliku od nastavnika, dobiva sve dozvole nad relacijom ispit. U upitima o ispitima mora koristiti relaciju ispit

```
SELECT * FROM ispit WHERE ocj = 1;
```

- kada korisnik novosel prestane biti dekan, ukinut će mu se dozvola nad relacijom ispit, a dodijeliti dozvola nad virtualnom relacijom ispitiZaNastavnike. U svojim upitima morat će koristiti virtualnu relaciju ispitiZaNastavnike

```
SELECT * FROM ispitiZaNastavnike WHERE ocj = 1;
```

# Upotreba sinonima

RJEŠENJE:

- Kreirati sinonime: alternativna imena za relacije ili virtualne relacije

korisnik s DBA dozvolom

```
CREATE PRIVATE SYNONYM kolar.ispitiZaSve FOR ispitiZaNastavnike;
CREATE PRIVATE SYNONYM ban.ispitiZaSve FOR ispitiZaNastavnike;
... sinonimi za ostale nastavnike i sinonim za dekana
CREATE PRIVATE SYNONYM novosel.ispitiZaSve FOR ispit;
```

- sada i dekan i nastavnici mogu koristiti isto ime objekta kada postavljaju upite o ispitima

```
SELECT * FROM ispitiZaSve WHERE ocj = 1;
```

- kada korisnik novosel prestane biti dekan

horvat

```
REVOKE SELECT, UPDATE ON ispit FROM novosel;
GRANT SELECT, UPDATE ON ispitiZaNastavnike TO novosel;
```

korisnik s DBA dozvolom

```
DROP SYNONYM novosel.ispitiZaSve;
CREATE PRIVATE SYNONYM novosel.ispitiZaSve FOR ispitiZaNastavnike;
```

- korisnik novosel će i dalje u svojim upitima moći koristiti ime objekta ispitiZaSve, ali će kao rezultat dobivati samo one podatke na koje, sada u svojstvu nastavnika, ima pravo

# Dodjeljivanje istih dozvola velikom broju korisnika

PROBLEM:

- svakom nastavniku treba dodijeliti dozvole za
  - pregled, unos i izmjenu podataka o ispitima za predmete koje predaje, pregled podataka iz relacije nast, iz relacije predaje, itd.
  - 150 nastavnika⇒150 puta treba obaviti niz naredbi za dodjelu dozvola:

```
GRANT SELECT, INSERT, UPDATE ON ispitiZaNastavnike TO kolar;
GRANT SELECT ON predmet TO kolar;
GRANT SELECT ON nast TO kolar;
...
-- ponoviti za svakog od 150 nastavnika
```

- za svakog novog zaposlenog nastavnika ponoviti postupak
- kada nastavnik ode u mirovinu, mora se obaviti niz REVOKE naredbi
- ako se promijene pravila pristupa (npr. odluči se da nastavnici mogu brisati "svoje" ispite), promjena se mora provesti za svakog nastavnika posebno:

```
GRANT DELETE ON ispitiZaNastavnike TO kolar;
-- ponoviti za svakog od 150 nastavnika
```

# Dodjeljivanje istih dozvola velikom broju korisnika

RJEŠENJE:

- definira se uloga (*role*), npr. nastavnik
- dozvole se, umjesto direktno korisnicima-nastavnicima, dodjeljuju ulozi

```
CREATE ROLE nastavnik;
GRANT SELECT, INSERT, UPDATE ON ispitiZaNastavnike TO nastavnik;
GRANT SELECT ON nast TO nastavnik;
GRANT SELECT ON predaje TO nastavnik;
...
```

- svakom nastavniku, umjesto cijelog niza dozvola, dovoljno je dodijeliti dozvolu za korištenje uloge nastavnik

```
GRANT nastavnik TO kolar;
GRANT nastavnik TO ban;
...
```

- ako nastavnik s identifikatorom korisnika ban ode u mirovinu

```
REVOKE nastavnik FROM ban;
```

- ako nastavnici trebaju dobiti dozvolu za brisanje "svojih" ispita

```
GRANT DELETE ON ispitiZaNastavnike TO nastavnik;
```

# Korištenje dozvola dobivenih putem uloga

- nakon uspostavljanja SQL-sjednice, korisnik posjeduje sljedeće dozvole:
  1. sve dozvole koje su dodijeljene PUBLIC "korisniku"
  2. sve dozvole koje su dodijeljene izravno dotičnom korisniku
  3. sve dozvole nad objektima kojima je dotični korisnik vlasnik
  4. dozvole na razini baze podataka (npr. ako korisnik ima DBA dozvolu, dopušteno mu je obavljanje svih operacija nad svim objektima)

- ako korisnik namjerava koristiti i dozvole dodijeljene nekoj ulozi, mora obaviti naredbu (npr.): `SET ROLE nastavnik;`
  - od tog trenutka, korisnik će (osim dozvola navedenih pod 1-4) imati i dozvole dodijeljene ulozi nastavnik.
- korisniku može biti dodijeljena više nego jedna uloga, ali u jednom trenutku može koristiti samo jednu od njih. Npr. nakon obavljanja naredbe:

```
SET ROLE studentskiSavjetnik;
```

  - korisnik će (osim dozvola navedenih pod 1-4) imati i dozvole dodijeljene ulozi studentskiSavjetnik (ali ne i ulozi nastavnik).
- naredbu `SET ROLE NONE;` korisnik koristi onda kad ne želi koristiti niti jednu ulogu

# Praćenje rada korisnika (*auditing*)

- evidentirati svaki pristup osjetljivim podacima u posebnoj datoteci za praćenje rada korisnika (*Audit Trail*)
- tipičan zapis datoteke sadrži sljedeće informacije:
  - SQL naredba koja se izvršava (*statement source*)
  - mjesto s kojeg je upućen zahtjev (terminal, IP adresa računala)
  - identifikator korisnika koji je pokrenuo operaciju
  - datum i vrijeme operacije
  - n-torke, atributi na koje se zahtjev odnosi
  - stara vrijednost n-torke
  - nova vrijednost n-torke
- sama činjenica da se prati "trag" obavljenih operacija nad podacima, često je dovoljna za sprečavanje zloporabe