Увод у релационе базе података

11б

Саша Малков
Универзитет у Београду
Математички факултет
2023/2024

[PM13]
Увод у РБП

Саша Малков

Тема 8

Безбедност база података

Безбедност база података

- Безбедност је стање ослобођености од опасности
- Са којим опасностима се суочавамо код база података?

Опасности

- Угрожавање поверљивости
- Угрожавање интегритета
- Угрожавање расположивости

Опасности

- Угрожавање поверљивости
  - неовлашћена лица имају приступ подацима

Опасности

- Угрожавање интегритета
  - угрожена је поузданост података
- Узроци
  - функционални проблеми рачунарског система
    - кварови, неисправности у софтверу (багови),...
  - неовлашћена лица су правила неисправне измене

Опасности

- Угрожавање расположивости
  - подаци нису на располагању
- Узроци
  - функционални проблеми рачунарског система
    - кварови, неисправности у софтверу (багови),...
  - неовлашћена онемогућавају приступ

Аспекти старања о безбедности

- Сигурност података
  - старање да само овлашћена лица могу да раде са подацима
  - данас се бавимо овим питањима
- Омогућавање опоравка
  - старање о резервним копијама података
  - старање о механизмима опоравка садржаја базе података
- За оба аспекта су надлежни најпре пројектанти, а затим администратори базе података

Сигурност података у РСУБП

- Сигурност података се остварује путем више концепата:
  - аутентикација
  - ауторизација
  - привилегије и власништво
  - енкрипција података
  - надзирање активности

Аутентикација

- Идентификовање корисника и проверавање идентитета
- Може да се остварује на два основна начина
  - коришћењем механизама оперативног система
  - коришћењем сопствених механизама СУБП
- У случају *DB2* користе се механизми оперативног система
  - у сложенијим окружењима може да се бира да ли се права проверавају на страни клијента, на страни сервера или помоћу посебних подсистема (на пример Керберос)

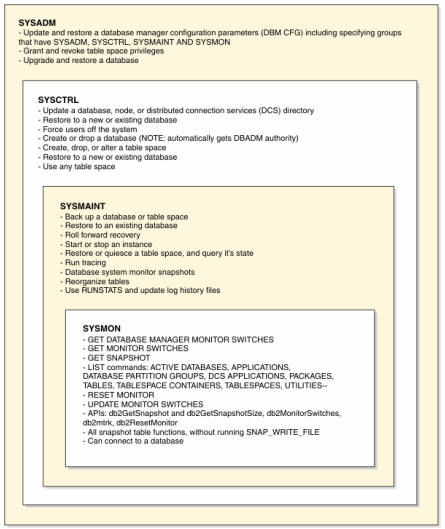
Ауторизација

- Ауторизација је одређивање каква права над којим објектима има аутентификовани корисник
  - права се додељују у облику дозвола - све што није дозвољено је аутоматски забрањено
- Ауторизација се одређује на два начина
  - давањем нивоа ауторизације
    - на нивоу СУБП
    - на нивоу конкретне базе података
  - додељивањем дозвола

Системски нивои ауторизације

- Системске ауторизације
  - *SYSADM* – системски администратор
    - сва права над свим ресурсима СУБП
    - укључује сва три преостала нивоа системских ауторизација
  - *SYSCTRL* – системски контролор
    - сва оперативна права над свим базама података
    - нема права приступања подацима
    - укључује *SYSMON*
  - *SYSMAINT* – одржавач система
    - има потребна права за извођење операција одржавања (промена конфигурационих параметара, прављење резервних копија и рестаурација на основу њих
    - нема права приступања подацима
    - укључује *SYSMON*
  - *SYSMON* – надзорник система
    - праћење функционисања система

Системски нивои ауторизације (*DB2*)



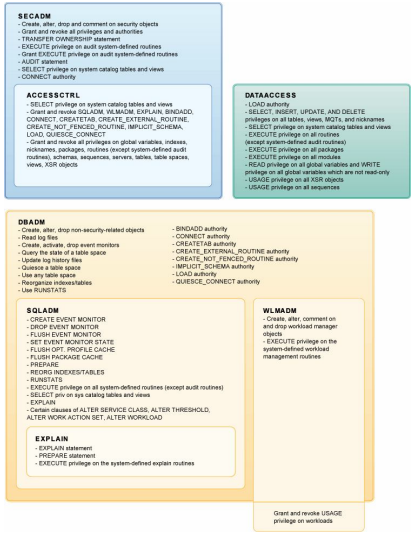
Нивои ауторизације на бази података

- Ауторизације на нивоу појединачне базе података
  - *DBADM* – администратор базе података
    - пуна административна права над свим објектима базе података
    - без права управљања безбедношћу (*SECADM*)
    - без права приступања подацима (*DATAACCESS*)
  - *SECADM* – администратор безбедности базе података
    - пуно управљање безбедносним елементима базе података
    - нема право приступања подацима
  - *ACCESSCTRL* – контролор приступа бази података
    - право управљања појединачним привилегијама на објектима базе података
  - *DATAACCESS* – приступ подацима
    - право пуног приступа свим подацима у бази података
  - …

Нивои ауторизације на бази података

- Ауторизације на нивоу појединачне базе података
  - …
  - *SQLADM* – *SQL* администратор
    - право надзора и оптимизовања рада *SQL* наредби у бази података
  - *WLMADM* – администратор оптерећења
    - управљање објектима за праћење оптерећења и управљање оптерећењем
  - *EXPLAIN* – контрола начина извршавања упита
    - право надзора и подешавања начина извршавања упита, али без приступа подацима
  - *LOAD* – масовно пуњење табела
    - право коришћења наредби за масовно пуњење табела базе података
  - *CONNECT* – корисник базе података
    - право повезивања на базу података

Нивои ауторизације на бази података (*DB2*)



Енкрипција

- Енкрипција се у базама података остварује на два основна нивоа
  - при складиштењу
    - заштита од физичке крађе података
  - при комуникацији
    - заштита података од прислушкивања
    - обично се користи *TLS* или сличан протокол
- Користе се и разни акцелератори
- Нећемо се тиме детаљније бавити

Надзирање активности

- Надзирање активности омогућава
  - откривање неуобичајених или осетљивих активности
  - накнадну анализу употребе
- Савремени системи надзора су врло флексибилни
  - дефинишу се политике
    - које операције се надзиру
    - на који начин
    - да ли само успешне или и неуспешне
  - бирају се објекти који се надзиру

Дозволе

- Дозволе се могу давати као:
  - примарне дозволе – оне које су експлицитно дате кориснику
  - секундарне дозволе – оне које су дате некој групи или улози а којима корисник припада
  - јавне дозволе – оне које су дате свима (“група” *PUBLIC*)
  - посебне дозволе – оне које се дају у посебним околностима
    - нећемо их овде разматрати
- Дозволе се дају
  - на нивоу објекта
    - које се дају за различите приступе конкретним објектима
  - на нивоу садржаја
    - практично се остварују путем погледа

SQL – DCL

- Рад са дозволама спада у део *SQL*-а који се назива *DCL* (*Data Control Language*)
- Основни елементи контролног дела *SQL*-а су
  - Наредба за додељивање дозвола – *GRANT*
  - Наредба за повлачење дозвола – *REVOKE*
  - Подсистем за рад са дозволама на погледима
  - Подсистем за рад са улогама

Додељивање дозвола

- Дозволе се додељују наредбом

  GRANT <lista-dozvola>
  ON <objekat>
  TO <lista-imena>
  [<opcije>]

- Скуп допуштених дозвола и начин навођења објекта зависе од врсте објекта
- У листи имена може да се нађе
  - корисничко име, облика: [USER] <ime>
  - име групе корисника, облика: [GROUP] <ime>
  - име улоге, облика: [ROLE] <ime>
  - скуп свих кориника, у облику: PUBLIC
- Допуштене опције зависе од врсте објекта

Додељивање дозвола (2)

- Упознаћемо рад са дозволама на најважнијим врстама објеката:
  - база података
  - схема
  - табела
  - поглед
  - индекс

Дозволе над базом података

- Дозволе се додељују наредбом

  GRANT <lista-dozvola> ON DATABASE TO <lista-imena>

- Листа дозвола обухвата ауторизације:
  - *DBADM*
  - *ACCESSCTRL*
  - *CONNECT*
  - …
- и дозволе:
  - *BINDADD* – право прављења пакета
  - *CREATETAB* – право прављења табела
  - …

Дозволе над табелама

- Дозволе се додељују наредбом

  GRANT <lista-dozvola> ON TABLE <ime-tabele> TO <lista-imena>
  [WITH GRANT OPTION]

- Дозволе могу да буду:
  - *ALL [PRIVILEGES]* – све дозволе осим *CONTROL*
  - *CONTROL* – све дозволе и право брисања табеле (и још понешто…)
  - *ALTER* – промена структуре табеле
  - *INSERT* – додавање редова
  - *UPDATE* [(<lista-imena-kolona->)]
  - *DELETE* – брисање редова
  - *SELECT* – читање садржаја
  - *INDEX* – прављење индекса
  - *REFERENCES* [(<lista-imena-kolona->)] – прављење страних кључева у односу на табелу
- Опција WITH GRANT OPTION – право да се наведена дозволе пренесу на трећа лица

Дозволе над погледима

- Дозволе се додељују наредбом
  GRANT <lista-dozvola> ON VIEW <ime-pogleda> TO <lista-imena>
  [WITH GRANT OPTION]
- Дозволе могу да буду:
  - *ALL [PRIVILEGES]* – све дозволе осим *CONTROL*
  - *CONTROL* – све дозволе и право брисања табеле (и још понешто...)
  - *INSERT* – додавање редова
  - *UPDATE* [(<lista-imena-kolona->)]
  - *DELETE* – брисање редова
  - *SELECT* – читање садржаја
- Опција WITH GRANT OPTION – право да се наведена дозволе пренесу на трећа лица

Дозволе над индексима

- Дозволе се додељују наредбом
  GRANT CONTROL ON INDEX <ime-indeksa> TO <lista-imena>
- Дозволе могу да буду:
  - *CONTROL* – дозвола да се обрише индекс

Повлачење дозвола

- Све додељене дозволе могу да се повуку
  REVOKE <lista-dozvola> ON <objekat> FROM <lista-imena>
- Важно:
  - *REVOKE ALL...* – повлачи све привилегије осим *CONTROL*
  - Одузимањем дозволе од некога ко је имао право преношења дозвола се ***не одузимају аутоматски*** дозволе које је он доделио
  - Не може да се одузме само право додељивања дозволе
    - мора да се одузме дозвола (а тиме и право додељивања)...
    - ...па да се додели без права додељивања
  - Не може да се одузме дозвола само за неке колоне
    - мора да се одузме...
    - ...па да се додели поново за мањи скуп колона

Схема као објекат БП

- Подсетимо се, схемом релационе базе података називамо скуп схема свих релација базе података
- Али исти термин се користи и као основа логичке организације
- Имена објеката базе података се организују у логичке целине које називамо схемама
  - пуна имена табела, погледа и индекса се састоје од имена схеме којој тај објекат припада и имена објекта
  - на пример, у бази података *STUD2020* имамо две схеме:
    - у схеми *DB* имамо скуп табела (и погледа и индекса) са *UTF8* садржајем
    - у схеми *DA* имамо скуп табела (и погледа и индекса) са *ASCII* садржајем

Специфичности имплементација

- У неким имплементацијама схема је само додатни описни атрибут објекта
  - на пример, у неким старијим имплементацијама система *SQL Server*, нису смеле да постоје две табеле које су у разчичитим схемама а имају исто име
- У неким имплементацијама схема је имала улогу базе података
  - на пример, неке старије имплементације система *Oracle* су биле конфигурисане тако да на једном серверу постоји тачно једна база података, али су схеме имале улогу не само логичког него и функционалног облика организације, налик на независне базе података
- Данас у скоро свим РСУБП схема има улогу логичке организације, налик фајл систему са једним нивоом директоријума

Схема и безбедност

- Схема није замишљена као безбедносни механизам
  - али основ логичког груписања може да буде безбедносна сличност
    - често се посебно осетљиви подаци групишу у одговарајуће схеме
    - олакшава се сагледавање привилегија на осетљивим објектима

Прављење схеме

- У систему *DB2* схема се прави наредбом:

```
CREATE SCHEMA
        [<schema-name>]                 (а)
        [AUTHORIZATION <name>]          (б)
        [<options>]
        [<commands>]
```

- Мора да се наведе бар једна од опција (а) и (б)
  - (а) прави се схема са датим именом, власник је онај ко је прави
  - (б) прави се схема са датим именом, власник је корисник са датим именом
  - (а,б) прави се схема са првим именом, а власнике је корисник са другим именом
- У склопу наредбе може да се наведе више наредби за прављење табела, погледа и индекса, као и за додељивање привилегија, у којима не мора да се наводи име схеме већ се подразумева да се ради у новој схеми
  - власник свих тако направљених објеката је власник схеме

Прављење схеме – пример

- На пример, наредном наредбом се
  - прави схема AAA
  - у њој се прави табела AAA.TAB1
  - у њој се прави и индекс AAA.IND1

```
CREATE SCHEMA aaa AUTHORIZATION bbb
CREATE TABLE tab1 (...)
CREATE INDEX ind1 ...
```

Имплицитно прављење схеме

- Ако се направи објекат у некој схеми, а да та схема није претходно направљена, онда се она аутоматски прави
  - власник тако направљене схеме је “систем”
    - у случају *DB2* то је “корисник” *SYSIBM*
- Ако се при прављењу објекта не наводи име, подразумева се да се прави у схеми чије име је идентично имену тренутног корисника

Брисање схеме

- За брисање схеме се користи наредба:
  DROP SCHEMA <name> RESTRICT
- Кључна реч *RESTRICT* наглашава да ће брисање схеме бити обустављено ако постоји макар један објекат у њој

Дозволе над схемама

- Дозволе се додељују наредбом
  GRANT <lista-dozvola> ON SCHEMA <ime-sheme> TO <lista-imena>
  [WITH GRANT OPTION]
- Дозволе могу да буду:
  - *ALTERIN* – дозвола да се мења структура свих објеката у схеми
  - *CREATEIN* – дозвола да се праве нови објекти у схеми
  - *DROPIN* – дозвола да се бришу сви објекти у схеми
- Опција WITH GRANT OPTION – право да се наведена дозволе пренесу на трећа лица

Погледи као безбедносни механизам

- Табеле не омогућавају додељивање дозвола за приступ појединачним редовима или колонама
  - осим у случају ажурирања или реферисања, где се дозвола може давати само за одређене колоне
- Ако је потребно да се приступ обезбеђује на основу садржаја, онда могу да се користе погледи

Погледи и безбедност – пример

- На пример, нека имамо табелу *DA.DOSIJE* и нека у студентској служби имамо службенике који раде само са личним подацима студената мастер студија
- Онда можемо да им омогућимо да раде на следећи начин
  - направимо одговарајући поглед *DA.DOSIJE_MASTER* који издваја само потребне колоне табеле и само редове који се односе на студенте мастер студија
  - забранимо тим корисницима да приступају табели *DA.DOSIJE*
  - дозволимо им да приступају погледу *DA.DOSIJE_MASTER*

Погледи и безбедност

- Важно:
  - да би поглед могао да се користи за додавање, мењање и брисање података, упит који га дефинише мора да буде над једном табелом, без колонским функција или израчунатих колона...
  - за безбедносна питања је веома важно да се користи опција провере услова
    - WITH CHECK OPTION
  - сложенији облици погледа могу да се користе уз дефинисање окидача

Улоге

- Важан елемент система сигурности су *улоге*
- Технички посматрано, концепт улоге је веома сличан концепту групе корисника
- Разлика је у томе што се, за разлику од група, улоге одржавају на нивоу СУБП
- Улога може да се
  - направи
  - да јој се додељују и одузимају дозволе (и улоге)
  - да јој се додељују и одузимају системске и ДБ ауторизације
  - да се корисницима додељују и одузимају улоге

Улоге (2)

- Улоге се праве према одговорностима и скуповима послова које нека врста корисника обавља
  - На пример:
    - улога “администратор студената” би имала сва права за рад са основним подацима о студентима
    - улога “администратор испита” би имала сва права за рад са подацима о испитима
    - улога “статистичар” би имала права за читање података потребних за статистичке извештаје, али можда не и појединачних личних података

Хијерархије улога

- Хијерархије улога настају када се једној улози додели друга
- Хијерархије улога могу да представљају
  - специјализације према пословима
    - нпр. *Наставник ->*
      - *Професор*
      - *Асистент*
  - специјализације према скуповима података
    - нпр. *Референт за сш.програме ->*
      - *Реф. за основне програме*
      - *Реф. за мастер програме*
      - *Реф. за докторске програме*

Разлике у односу на групе

- Код неких СУБП постојје ограничења у односу на групе (разликују се према конкретном СУБП)
- На пример, код *DB2*
  - при прављењу неких врсте објеката (пакети, погледи, окидачи,...), не проверавају се права по групама али се проверавају по улогама
  - улоге не могу да буду власници објеката (групе у неким случајевима могу)

Прављење улоге

- Улога се прави наредбом

  CREATE ROLE <ime uloge>
- На пример:

  CREATE ROLE nastavnik
- Улоге се праве на нивоу СУБП, па може да их прави само корисник са ауторизацијом *SECADM*

Брисање улоге

- Улога се брише наредбом

  DROP ROLE <ime uloge>

Дозволе по улогама

- Дозволе се додељују и одузимају улогама као и корисницима и групама
  - *GRANT*
  - *REVOKE*

Додељивање улоге

- Улоге се додељују на сличан начин као и дозволе:

```
GRANT [ ROLE ] <lista-uloga>
  TO <lista-imena>
  [ WITH ADMIN OPTION ]
```

- Међу именима којима се додељује улога могу да буду корисници, групе, улоге и *PUBLIC*
- Опција *WITH ADMIN OPTION* наглашава да се додељује и право преношења улоге на друге

Одузимање улоге

- Улоге се одузимају на сличан начин као и дозволе:

```
REVOKE [ADMIN OPTION FOR] [ ROLE ] <lista-uloga>
  FROM <lista-imena>
  [ BY ALL ]
```

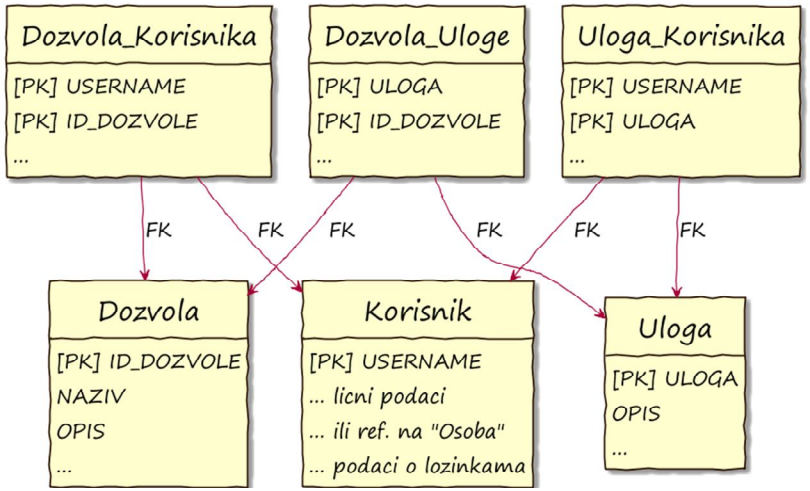
- Међу именима којима се одузима улога могу да буду корисници, групе, улоге и *PUBLIC*
- Ако се наведе опција *ADMIN OPTION FOR* онда се одузима само право преношења улоге на друге за наведене улоге, али не и саме улоге
- Опција *BY ALL* само наглашава да ће улога бити одузета без обзира на то ко ју је доделио – ако се не наведе, понашање је неизмењено

Апликативне привилегије

- При развоју апликација (и информационих система) обично не могу да се непосредно примене системи привилегија које подржава СУБП
  - апликација прави свој подсистем за рад са корисницима
    - корисници се моделирају као ентитети у бази података
    - морају да имају различита права над различитим деловима апликације
    - апликација сама проверава њихова права током рада
  - апликација се у односу на базу података обично пријављује као посебан *апликативни* корисник

Подсистем за рад са корисницима

- Подсистем за рад са корисницима се обично имплементира тако да подражава рад са улогама СУБП
  - Дефинишу се
    - привилегије (дозволе)
    - корисници
    - улоге
  - Омогући се давање привилегија на нивоу корисника и улога
  - Омогући се додељивање улога корисницима



Апликативни налози

- Апликација се у односу на базу података обично пријављује као посебан *апликативни* корисник
  - То значи да база података *не зна* ко је стварно пријављени корисник, т.ј. не може непосредно да га повеже са апликативним системом привилегија
- Већина савремених СУБП омогућава апликацији да при пријављивању корисника постављањем посебних регистара *упозна* базу података са апликатинвим корисником

Апликативни налози (2)

- Код *DB2*, након аутентикације корисника могу да се поставе два посебна регистра
  - CLIENT_APPLNAME
    - назив активне апликације
  - CLIENT_USERID
    - корисничко име у апликацији
  - из сигурносних разлога не могу да се поставе кроз *SQL*
- Затим ти подаци могу да се користе у раду током сесије
  - експлицитно и имплицитно у упитима
    - на пример, тако да се резултати упита разликују према кориснику
  - имплицитно у окидачима
    - на пример, тако да се се аутоматски евидентира ко је када које податке уносио и кроз коју апликацију

Експлицитни журнал

- Понекад има смисла да се експлицитно имплементира прављење дневника активности помоћу окидача
- На пример, када се промени податак о предмету, можемо да запишемо ко је и када то променио


```
create trigger journal.PREDMET_UPDATE
    after update on DB.PREDMET
    referencing new_table as new
    for each statement
    mode db2sql
        insert into journal.PREDMET(
                ID, OZNAKA, NAZIV, ESPB,
                journal_time,
                journal_app,
                journal_user,
                journal_action
        )
        select
                ID, OZNAKA, NAZIV, ESPB,
                current_timestamp,
                client applname,
                client userid,
                'U'
        from new
```


Литература за тему

- Гордана Павловић-Лажетић, **Увод у релационе базе података,** *2.изд. Математички факултет, 1999.*
  - доступно онлајн: *http://poincare.matf.bg.ac.rs/~gordana//urbp-2016.htm*
- Документација за *DB2 11.5*:
  - онлајн:
    - https://www.ibm.com/support/knowledgecenter/SSEPGG_11.5.0
  - ПДФ:
    - https://www.ibm.com/support/pages/node/627743