Napredne baze podataka

Dozvole

Tehničko veleučilište u Zagrebu
Informatičko-računarski odjel

Sadržaj predavanja

- Sigurnost – integritet podataka
- Zaštita podataka
- Dodjeljivanje dozvola

Sigurnost - Integritet

Security - Integrity

- Sigurnost
  - osigurava da su korisnici ovlašteni za akcije koje pokušavaju izvesti
- Integritet
  - osigurava da su akcije koje korisnici pokušavaju izvesti ispravne
- U oba slučaja:
  - moraju biti definirana pravila koja korisnici ne smiju narušiti
  - pravila se pohranjuju u rječnik podataka
  - DBMS nadgleda rad korisnika – osigurava poštivanje pravila

Sigurnost baze podataka

Database security

- sprječavanje pristupa neovlaštenim osobama
  - **zaštita na razini operacijskog sustava**
  - **zaštita na razini DBMS-a**
- definiranje pravila pristupa (security rules, authorization rules)
- šifriranje podataka

•Problem zaštite podataka

- zakonski, socijalni i etički aspekt
  - ima li operater zakonsko pravo na korisnikove informacije - stanje računa, iznos plaće
- strategijski aspekt
  - tko definira pravila pristupa – tko određuje kakve ovlasti ima pojedini korisnik
- operativni aspekt
  - kako osigurati poštivanje pravila – kojim se mehanizmima osigurava poštivanje definiranih pravila
  - kako su zaštićene lozinke, koliko se često mijenjaju

Ustav RH - Članak 37.
Zakon o zaštiti osobnih podataka

- 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.

•Način zaštite podataka

- diskrecijski pristup
  - korisnik ima različita prava pristupa (privilege, authority) različitim objektima
  - različiti korisnici imaju različita prava nad istim objektima

- mandatni pristup
  - svaki objekt ima oznaku klasifikacijske razine
    - (classification level)
  - svakom korisniku dodijeljena je oznaka razine ovlasti
    - (clearance level)
  - neki objekt dostupan je korisnicima koji imaju odgovarajuću razinu dozvola

Dozvole pristupa podacima

- Korisnik dozvole
  - korisnik s određenom identifikacijskom oznakom (userID)
  - bilo koji korisnik - PUBLIC
- Objekt
  - baza podataka
  - tablica (relacija)
  - stupac tablice (atribut)
  - skup n-torki iz tablice
  - izvedena tablica (pogled, virtualna tablica)
  - pohranjena procedura

MySQL baza

- Promjenu ovlasti moguće je napraviti
  - Kroz legalni SQL način rada (GRANT)
  - Kroz promjene u bazi imena '*MySQL*' koja se mora nalaziti na poslužitelju
- Poslije provedenih promjena mora se (ovisno):
  - Ponovno pokrenuti servis
  - Zadati komandu FLUSH PRIVILEGES iz komandnog moda

•Stvaranje korisnika

```
CREATE USER user [IDENTIFIED BY [PASSWORD] 'password']
    [, user [IDENTIFIED BY [PASSWORD] 'password']] ...
```

- ***user*** ->mora biti oblika **'korisničko_ime'@'adresa_klijenta'**
- ako se ne definira @'adresa_klijenta', pretpostavlja se adresa **%**
- **%** -> označava da se korisnik može spojiti sa svih adresa (ne provjerava se)

- naredba CREATE USER dodat će redak u *mysql.user*

Dodjeljivanje dozvola korisniku

- Naredba: **GRANT**
- Ako se pokreće GRANT naredba, onda nije potrebno posebno stvarati korisnika naredbom CREATE USER
- S obzirom da se dozvole dodjeljuju naredbom GRANT, kod brisanja korisnika treba biti pažljiv:
  - Prvo sa **SHOW GRANT** provjeriti koje dozvole ima taj korisnik
  - Sa **REVOKE** naredbom skinuti dozvole s korisnika
  - U zadnjem koraku obrisati korisnika naredbom **DROP USER**

•GRANT

```
GRANT
    priv_type [(column_list)]
      [, priv_type [(column_list)]] ...
    ON [object_type]
        {
            *
          | *.*
          | db_name.*
          | db_name.tbl_name
          | tbl_name
          | db_name.routine_name

        }
    TO user [IDENTIFIED BY [PASSWORD] 'password']
        [, user [IDENTIFIED BY [PASSWORD] 'password']] ...
    [REQUIRE
        NONE |
        [{SSL| X509}]
        [CIPHER 'cipher' [AND]]
        [ISSUER 'issuer' [AND]]
        [SUBJECT 'subject']]
    [WITH with_option [with_option] ...]
```

```
object_type =
      TABLE
    | FUNCTION
    | PROCEDURE

with_option =
      GRANT OPTION
    | MAX_QUERIES_PER_HOUR count
    | MAX_UPDATES_PER_HOUR count
    | MAX_CONNECTIONS_PER_HOUR count
    | MAX_USER_CONNECTIONS count
```

•Primjeri GRANT

```
GRANT ALL ON *.* TO 'someuser'@'somehost';
```

Za korisnika koji se spaja (@) s klijenta

Tip privilegije
ALL-svi
SELECT
INSERT
DROP
UPDATE
….

Razina dozvola
Npr: *.* -sve baze i sve tablice
Npr: *autoservis.** -> sve tablice iz baze autoservis

http://http://dev.mysql.com/doc/refman/5.6/en/grant.html

•Primjeri GRANT

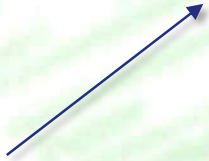
```
GRANT ALL ON *.* TO 'someuser'@'somehost'
IDENTIFIED BY 'lozinka' REQUIRE SSL;
```

Zahtjevani tip spoja (nije obavezno)

Lozinka koja se koristi za spajanje
(nije obavezno ako je prije stvoren korisnik)

•Primjeri GRANT

```
GRANT ALL ON *.* TO 'someuser'@'somehost'
IDENTIFIED BY 'lozinka' WITH GRANT OPTION;
```



Korisnik može dalje izdavati lozinke drugima !

```
GRANT ALL ON *.* TO 'someuser'@'somehost'
IDENTIFIED BY 'lozinka' WITH
 MAX_QUERIES_PER_HOUR 100;
```

Zadani limiti!

Razine Grant

- pristup sustavu
  - Tablica *mysql.user*
  - `GRANT ALL ON *.* TO 'someuser'@'somehost';`
  - `GRANT SELECT, INSERT ON *.* TO 'someuser'@'somehost';`
- pristup bazi
  - Tablica *mysql.db*
  - `GRANT ALL ON mydb.* TO 'someuser'@'somehost';`
  - `GRANT SELECT, INSERT ON mydb.* TO 'someuser'@'somehost';`

•Razine Grant

- Pristup tablici
  - Tablica mysql.tables_priv
  - `GRANT ALL ON mydb.mytbl TO 'someuser'@'somehost';`
  - `GRANT SELECT, INSERT ON mydb.mytbl TO 'someuser'@'somehost';`
- Pristup atributu
  - Tablica *mysql.columns_priv*
  - `GRANT SELECT (col1), INSERT (col1,col2) ON mydb.mytbl TO 'someuser'@'somehost';`
- Pristup razini / pohranjenom zadatku
  - Tablica *mysql.proc*
  - `GRANT CREATE ROUTINE ON mydb.* TO 'someuser'@'somehost';`
  - `GRANT EXECUTE ON PROCEDURE mydb.myproc TO 'someuser'@'somehost';`

•Revoke

```
REVOKE
    priv_type [(column_list)]
      [, priv_type [(column_list)]] ...
    ON [object_type]
        {
            *
          | *.*
          | db_name.*
          | db_name.tbl_name
          | tbl_name
          | db_name.routine_name

        }
    FROM user [, user] ...

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

•Revoke

- Sintaksa je slična kao za grant
- Iznimno naredba

  `REVOKE ALL PRIVILEGES, GRANT OPTION FROM user;`

  briše sve privilegije koje je korisnik dobio
- Kao i kod GRANT, promjene se neće primijetiti *
  - Dok se ponovno ne pokrene baza podataka
  - Ili zada komanda `FLUSH PRIVILEGIES;`
- Ukidanje dozvola možemo obaviti i brisanjem zapisa u bazi Mysql

- `REVOKE SELECT ON mydb.* FROM 'user'@'localhost';`

•Rename

```
RENAME USER old_user TO new_user
    [, old_user TO new_user] ...
```

- Mijenja ime korisnika
- Mijenja podatke u tablici *mysql.user*

- NAGLASAK:
  - MySQL neće promijeniti ostale zapise koji se odnose na stvorene baze od tog korisnika ni njegove dozvole
  - Navedene dozvole treba dodijeliti ponovno
  - Lakše je koristiti REVOKE pa ponovno GRANT

•Postavljanje lozinke

```
SET PASSWORD [FOR user] =
    {
        PASSWORD('some password')
      | OLD_PASSWORD('some password')
      | 'encrypted password'
    }
```

- Postavlja lozinku za korisnika
  - `SET PASSWORD FOR 'bob'@'%.loc.gov' = PASSWORD('newpass');`
- Isto kao i promjena u tablici *mysql.user*
  - `UPDATE mysql.user SET Password=PASSWORD('newpass') WHERE User='bob' AND Host='%.loc.gov';`
  - `FLUSH PRIVILEGES;`

•CHECK TABLE

```
CHECK TABLE tbl_name [, tbl_name] ... [option] ...

option = {FOR UPGRADE | QUICK | FAST | MEDIUM | EXTENDED | CHANGED}
```

- Provjerava ispravnost tablice
  - QUICK – provjerava postojanost redova (krivi linkovi)
  - FAST – samo je li tablica ispravno zatvorena
  - CHANGED – provjerava jesu li tablice koje su imale promjenu ispravno zatvorene
  - MEDIUM – provjerava svaki red i obrisane linkove
  - EXTENDED – detaljno provjerava svaki unos

•OPTIMIZE TABLE

```
OPTIMIZE [LOCAL | NO_WRITE_TO_BINLOG] TABLE tbl_name [, tbl_name] ...
```

- Ako je obrisan veliki dio tablice
- Naročito ako se radi o tablici s BLOB podacima
- Optimize
  - Popravlja nedostatke u tablici nastale uslijed brisanja
  - Popravlja indekse
  - Popravlja statistiku tablice

•REPAIR TABLE

```
REPAIR [LOCAL | NO_WRITE_TO_BINLOG] TABLE
    tbl_name [, tbl_name] ... [QUICK] [EXTENDED] [USE_FRM]
```

- Popravlja tablicu
  - QUICK – popravlja samo stablo indeksa
  - XTENDED – obnavlja cijelu tablicu
  - FROM – detalj je li se koristi dodatna datoteka koja se koristi kod pohrane tablice (specifično za MySQL)

•SHOW komande

```
SHOW AUTHORS
SHOW CHARACTER SET [like_or_where]
SHOW COLLATION [like_or_where]
SHOW [FULL] COLUMNS FROM tbl_name [FROM db_name] [like_or_where]
SHOW CONTRIBUTORS
SHOW CREATE DATABASE db_name
SHOW CREATE EVENT event_name
SHOW CREATE FUNCTION funcname
SHOW CREATE PROCEDURE procname
SHOW CREATE TABLE tbl_name
SHOW CREATE TRIGGER trigger_name
SHOW CREATE VIEW view_name
SHOW DATABASES [like_or_where]
SHOW ENGINE engine_name {STATUS | MUTEX}
SHOW [STORAGE] ENGINES
SHOW ERRORS [LIMIT [offset,] row_count]
SHOW [FULL] EVENTS
SHOW FUNCTION CODE sp_name
SHOW FUNCTION STATUS [like_or_where]
SHOW GRANTS FOR user
SHOW INDEX FROM tbl_name [FROM db_name]
SHOW INNODB STATUS
SHOW OPEN TABLES [FROM db_name] [like_or_where]
```

```
SHOW PLUGINS
SHOW PROCEDURE CODE sp_name
SHOW PROCEDURE STATUS [like_or_where]
SHOW PRIVILEGES
SHOW [FULL] PROCESSLIST
SHOW SCHEDULER STATUS
SHOW [GLOBAL | SESSION] STATUS [like_or_where]
SHOW TABLE STATUS [FROM db_name] [like_or_where]
SHOW TABLES [FROM db_name] [like_or_where]
SHOW TRIGGERS [FROM db_name] [like_or_where]
SHOW [GLOBAL | SESSION] VARIABLES [like_or_where]
SHOW WARNINGS [LIMIT [offset,] row_count]

like_or_where:
    LIKE 'pattern'
  | WHERE expr
```

•SHOW

- SHOW binary logs
  - Vraća brojeve binarnih logova koji se trenutno koriste
- SHOW character set
  - Pokazuje podržane setove znakova
- SHOW collation
  - Pokazuje podržane kolotacije znakova
- SHOW COLUMNS FROM tablica
  - Opisuje relacijsku shemu
- SHOW CREATE DATABASE
  - Pokazuje naredbu koja je stvorila bazu
- SHOW CREATE SHEMA
  - Pokazuje naredbu koja je stvorila relacijsku shemu

•SHOW

- SHOW CREATE event
  - Pokazuje koji je događaj potreban za pokretanje
- SHOW CREATE PROCEDURE i SHOW CREATE FUNCTION
  - Opisuje procedure i funkcije zadane u bazi
- SHOW CREATE TABLE
  - Pokazuje naredbu koja je kreirala tablicu
- SHOW CREATE TRIGGER
  - Pokazuje naredbu koja je kreirala okidač
- SHOW CREATE VIEW
  - Pokazuje naredbu koja je kreirala pogled
- ...

- http://dev.mysql.com/doc/refman/5.6/en/show.html