

Securitatea unei baze de date

Asigurarea integritatii si consistentei  
datelor

# SQL – Securitatea unei baze de date

Politica de securitate a unei baze de date presupune definirea de conturi de utilizare si atribuirea de drepturi de acces acestor utilizatori. Fiecarui utilizator in parte i se pot acorda drepturi pentru accesul (consultarea), modificarea, stergerea datelor in mod individual. Accesul poate fi restrictionat pentru tipuri de structuri de date, pentru anume structuri de date (baze de date, tabele, campuri ale unor tabele) pentru crearea si modificarea de proceduri. Cateva reguli simple pentru asigurarea securitatii unei baze de date:

- Intotdeauna protejati user-ul root cu parola;
- Nu acordati drept de acces la tabela user din mysql altui user decat root;
- Nu stocati parole in text clar in baza de date, folositi intotdeauna MD5() sau SHA1() sau alta metoda de acelasi fel si stocati rezultatul incryptarii;
- Nu acordati niciodata drepturi tuturor utilizatorilor;
- Nu folositi un dictionar de parole pentru alegerea parolelor;
- Limitati accesul la masina host; intotdeauna acordati fiecarui utilizator minimum de drepturi necesare pentru utilizarea aplicatiei, in acord cu logica de business a aplicatiei;
- Drepturile acordate utilizatorilor cu ajutorul comenzii GRANT pot fi revocate utilizand comanda REVOKE; nu ezitati sa o folositi pentru a va asigura ca nivelul de securitate necesar este pastrat!

# SQL – Securitatea unei baze de date

## Crearea unui utilizator

```
CREATE USER user_specification
    [, user_specification] ...

user_specification:
    user
    [
        IDENTIFIED BY [PASSWORD] 'password'
    | IDENTIFIED WITH auth_plugin [AS 'auth_string']
    ]
```

#Creeaza un utilizator 'adrian' cu parola 'a12345'

```
CREATE USER 'adrian'@'localhost'
IDENTIFIED BY 'a12345';
```

#Creeaza un utilizator 'daniel' care se poate conecta fara parola (nu este sigur):

```
CREATE USER 'daniel'@'localhost';
```

#Creeaza un utilizator 'gabriel' cu parola (criptata) '\*90E462C37378CED12064BB3388827D2BA3A9B689'

```
CREATE USER 'gabriel'@'localhost'
IDENTIFIED BY PASSWORD '*90E462C37378CED12064BB3388827D2BA3A9B689';
```

#Verificam incriptarea unei parole: ex. cea salvata pentru daniel: 'a12345';

```
select user, password from user where password('a12345') = password;
```

# SQL – Securitatea unei baze de date

```
#Sterge utilizator 'adrian'  
DROP USER 'adrian'@'localhost';  
  
#Redenumeste un utilizator  
RENAME USER daniel TO cristi;
```

```
c:\wamp\bin\mysql\mysql5.5.24\bin\mysql.exe  
mysql> drop user adrian@localhost;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> select user, password from user;  
+-----+-----+  
| user      | password  
+-----+-----+  
| root      | *117AFDB7BD4670EE99D5F286C03ABE2588427856  
| root      | *117AFDB7BD4670EE99D5F286C03ABE2588427856  
| root      | *117AFDB7BD4670EE99D5F286C03ABE2588427856  
| preda     | *00BE18440B3AA4133A09DD4EC89520B610895210  
| daniel     | *90E462C37378CED12064BB3388827D2BA3A9B689  
| gabriel   | *90E462C37378CED12064BB3388827D2BA3A9B689  
+-----+-----+  
6 rows in set (0.00 sec)  
  
mysql>
```

```
c:\wamp\bin\mysql\mysql5.5.24\bin\mysql.exe  
mysql> RENAME USER daniel TO cristi;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> select user, password from user;  
+-----+-----+  
| user      | password  
+-----+-----+  
| root      | *117AFDB7BD4670EE99D5F286C03ABE2588427856  
| root      | *117AFDB7BD4670EE99D5F286C03ABE2588427856  
| root      | *117AFDB7BD4670EE99D5F286C03ABE2588427856  
| preda     | *00BE18440B3AA4133A09DD4EC89520B610895210  
| cristi    | *90E462C37378CED12064BB3388827D2BA3A9B689  
| gabriel   | *90E462C37378CED12064BB3388827D2BA3A9B689  
+-----+-----+  
6 rows in set (0.00 sec)  
  
mysql>
```

# SQL – Securitatea unei baze de date

```
c:\wamp\bin\mysql\mysql5.5.24\bin\mysql.exe

mysql> select user, password from user;

+----+-----+
| user      | password                                     |
+----+-----+
| root      | *117AFDB7BD4670EE99D5F286C03ABE2588427856 |
| root      | *117AFDB7BD4670EE99D5F286C03ABE2588427856 |
| root      | *117AFDB7BD4670EE99D5F286C03ABE2588427856 |
| preda     | *00BE18440B3AA4133A09DD4EC89520B610895210 |
| adrian    | *9B09A5B43380B18B737EDE4E457F22A8E95AB82B |
| daniel    |                                             |
| gabriel   | *90E462C37378CED12064BB3388827D2BA3A9B689 |
+----+-----+

7 rows in set (0.00 sec)

mysql>
```

```
c:\wamp\bin\mysql\mysql5.5.24\bin\mysql.exe

mysql> select user, password from user where password('a12345') = password;

+----+-----+
| user      | password                                     |
+----+-----+
| adrian    | *9B09A5B43380B18B737EDE4E457F22A8E95AB82B |
+----+-----+

1 row in set (0.00 sec)

mysql>
```

Stergerea unui utilizator

```
DROP USER user [, user] ...
```

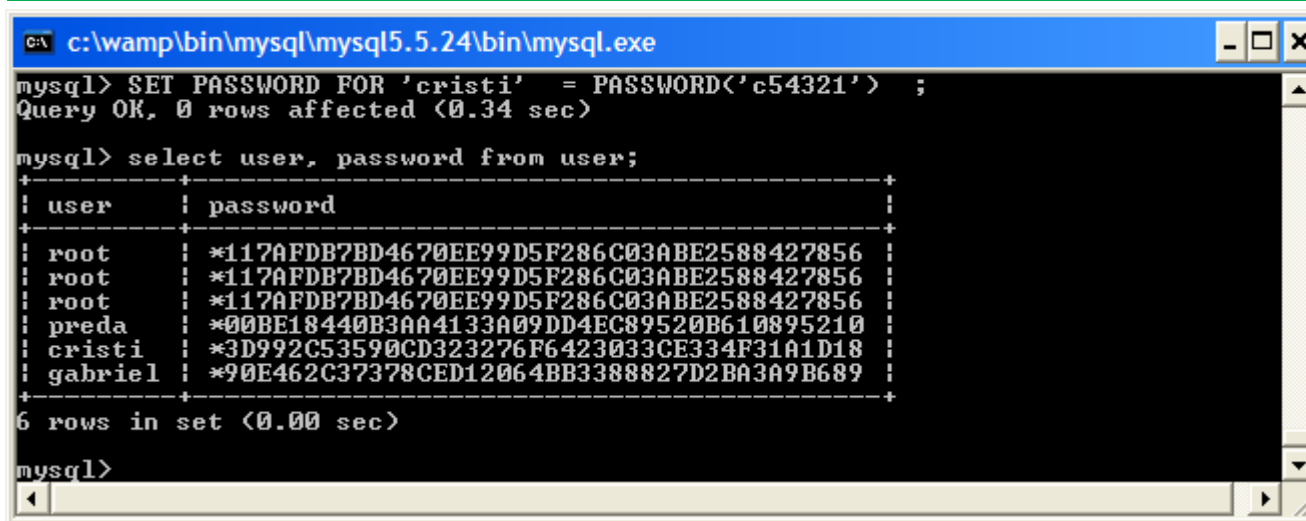
Redenumirea unui utilizator

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

# SQL – Securitatea unei baze de date

## Setarea unei parole

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



```
c:\wamp\bin\mysql\mysql5.5.24\bin\mysql.exe  
mysql> SET PASSWORD FOR 'crisi' = PASSWORD('c54321') ;  
Query OK, 0 rows affected (0.34 sec)  
  
mysql> select user, password from user;  
+-----+-----+  
| user      | password |  
+-----+-----+  
| root      | *117AFDB7BD4670EE99D5F286C03ABE2588427856 |  
| root      | *117AFDB7BD4670EE99D5F286C03ABE2588427856 |  
| root      | *117AFDB7BD4670EE99D5F286C03ABE2588427856 |  
| preda     | *00BE18440B3AA4133A09DD4EC89520B610895210 |  
| crisi     | *3D992C53590CD323276F6423033CE334F31A1D18 |  
| gabriel   | *90E462C37378CED12064BB3388827D2BA3A9B689 |  
+-----+-----+  
6 rows in set (0.00 sec)  
  
mysql>
```

### #Seteaza parola

```
SET PASSWORD FOR 'adrian'@'localhost' = PASSWORD('pass12345');
```

### #Este echivalent cu:

```
UPDATE mysql.user SET Password=PASSWORD('pass12345')  
WHERE User='adrian' AND Host='localhost';  
FLUSH PRIVILEGES;
```

# SQL – Securitatea unei baze de date

## Acordarea de privilegii folosind comanda GRANT

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

GRANT PROXY ON user_specification
    TO user_specification [, user_specification] ...
    [WITH GRANT OPTION]

object_type:
    TABLE
    | FUNCTION
    | PROCEDURE

priv_level:
    *
    | *.*
    | db_name.*
    | db_name.tbl_name
    | tbl_name
    | db_name.routine_name
```

# SQL – Securitatea unei baze de date

## Acordarea de privilegii folosind comanda GRANT (continuare)

```
user_specification:  
  user  
  [  
    IDENTIFIED BY [PASSWORD] 'password'  
    | IDENTIFIED WITH auth_plugin [AS 'auth_string']  
  ]  
  
ssl_option:  
  SSL  
  | X509  
  | CIPHER 'cipher'  
  | 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_CONNECTIONS count
```



# SQL – Securitatea unei baze de date

## Acordarea de privilegii folosind comanda GRANT: exemple

```
#Acorda privilegii
#Acorda privilegii lui 'gabriel' pentru a crea o baza de date gabriel
GRANT CREATE ON gabriel TO 'gabriel'@'localhost';

#Acorda privilegii lui 'gabriel' pentru operatiunea SELECT pentru gabriel
GRANT SELECT ON gabriel.* TO 'gabriel'@'localhost';

#Acorda privilegii lui 'gabriel' pentru operatiunea SELECT pentru baza de date gabriel, tabela report
GRANT SELECT ON gabriel.report TO 'gabriel'@'localhost';

#Acorda privilegii lui 'adrian' pentru operatiunile DELETE si DROP pentru baza de date gabriel, tabela report
GRANT DELETE, DROP ON gabriel.report TO 'gabriel'@'localhost';

#Acorda drepturile de a folosi un numar de 20 de ori pe ora utilizatorului 'gabriel'
GRANT USAGE ON *.* TO 'gabriel'@'localhost' WITH MAX_QUERIES_PER_HOUR 20;
# Toate privilegiile pentru baza de date 'gabriel'
GRANT ALL ON gabriel.* TO 'gabriel'@'localhost';
```

# SQL – Securitatea unei baze de date

## Revocarea privilegiilor

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

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

REVOKE PROXY ON user
  FROM user [, user] ...
```

```
#Elimina privilegii
#Elimina drepturile de a crea si drop pentru 'gabriel'
REVOKE CREATE, DROP ON gabriel.* FROM 'gabriel'@'localhost';

#Elimina toate privilegiile lui gabriel
REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'gabriel';
```

# SQL – Asigurarea integritatii si consistentei datelor

Asigurarea integritatii si consistentei datelor presupune:

- Integritatea entitatii: aceasta se asigura prin impunerea existentei unei chei primare; de asemenea, o cheie primara nu poate avea valoarea NULL; in acest fel se asigura unicitatea unei inregistrari;
- Integritatea referentiala: aceasta presupune mentinerea relatiilor intre inregistrarile din tabele diferite; se asigura prin impunerea mecanismului de cheie straina (din una din tabele) care face referinta la o cheie primara (din alta tabela).
- Integritatea domeniului: orice camp are o anumita dimensiune sau plaja de valori care corespunde tipului de date folosite pentru acel camp; aplicatia va impune respectarea acestui domeniu de definitie sau dimensiune pentru fiecare camp al fiecărei tabele;
- Integritatea conform definitiei utilizatorului: pentru unele campuri se pot impune (conform logicii de business a aplicatiei) restrictii care trebuie respectate.

# SQL – Asigurarea integritatii si consistentei datelor

## Integritatea referentiala

```
#Creare tabela adresa
CREATE TABLE IF NOT EXISTS adresa
(id int unique auto_increment primary key,
strada char(20),
oras char(20));

#Creare tabela client; client defineste o cheie straina adresa_id care are referinta catre campul id din adresa
CREATE TABLE IF NOT EXISTS client
(id int unique auto_increment primary key,
denumire char(50),
adresa_id int,
INDEX (adresa_id),
FOREIGN KEY (adresa_id) REFERENCES adresa(id));
```

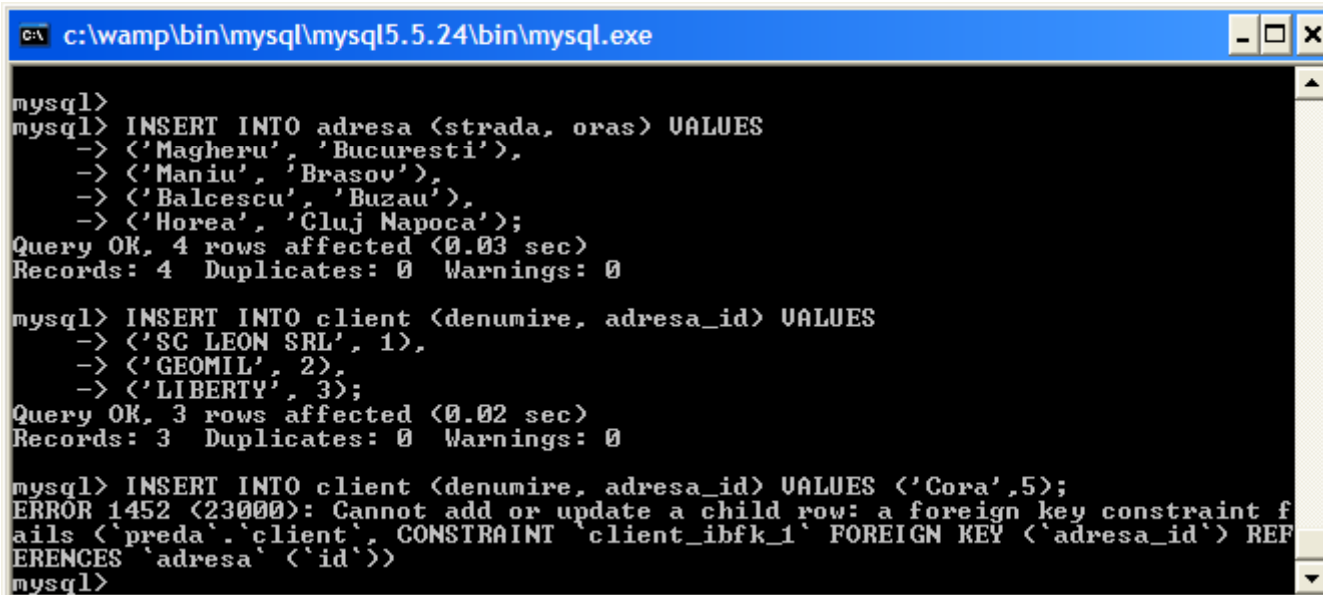
```
# Adaugam 4 inregistrari in tabela de adrese
INSERT INTO adresa (strada, oras) VALUES
('Magheru', 'Bucuresti'),
('Maniu', 'Brasov'),
('Balcescu', 'Buzau'),
('Horea', 'Cluj Napoca');

#adaugam 3 clienti, cu id-ul maxim 3 (in tabela de adrese, id-ul max este 4)
INSERT INTO client (denumire, adresa_id) VALUES
('SC LEON SRL', 1),
('GEOMIL', 2),
('LIBERTY', 3);
```

# SQL – Asigurarea integritatii si consistentei datelor

Incercarea de a adauga o inregistrare avand valoarea adresa\_id inexistentă in multimea valorilor existente pentru id din tabela adresa provoacă o eroare: "cannot add or update a child row: a foreign key constraint fails"

```
# Incercam sa adaugam o noua inregistrare in tabela de clienti, cu o valoare pentru adresa_id  
# care nu se gaseste intre valorile lui id din tabela de adrese; incercarea va esua  
INSERT INTO client (denumire, adresa_id) VALUES ('Cora',5);
```



```
c:\wamp\bin\mysql\mysql5.5.24\bin\mysql.exe

mysql>
mysql> INSERT INTO adresa (strada, oras) VALUES
-> ('Magheru', 'Bucuresti'),
-> ('Maniu', 'Brasov'),
-> ('Balcescu', 'Buzau'),
-> ('Horea', 'Cluj Napoca');
Query OK, 4 rows affected (0.03 sec)
Records: 4 Duplicates: 0 Warnings: 0

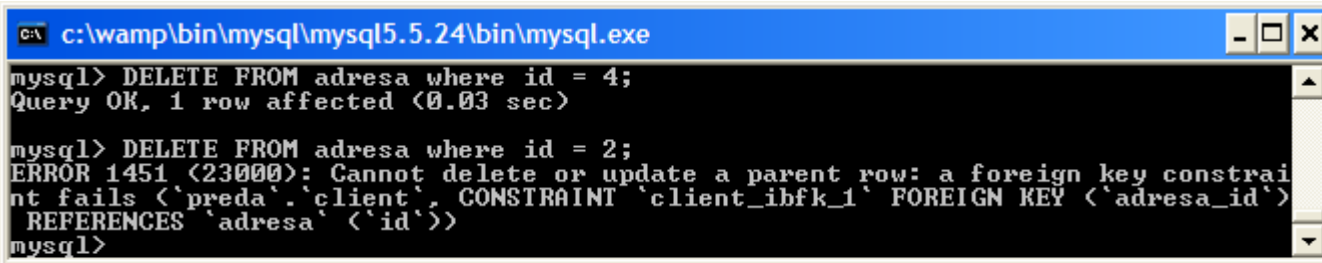
mysql> INSERT INTO client (denumire, adresa_id) VALUES
-> ('SC LEON SRL', 1),
-> ('GEOMIL', 2),
-> ('LIBERTY', 3);
Query OK, 3 rows affected (0.02 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> INSERT INTO client (denumire, adresa_id) VALUES ('Cora',5);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint f
ails ('preda`.`client`, CONSTRAINT `client_ibfk_1` FOREIGN KEY (`adresa_id`) REF
ERENCES `adresa` (`id`))
mysql>
```

# SQL – Asigurarea integritatii si consistentei datelor

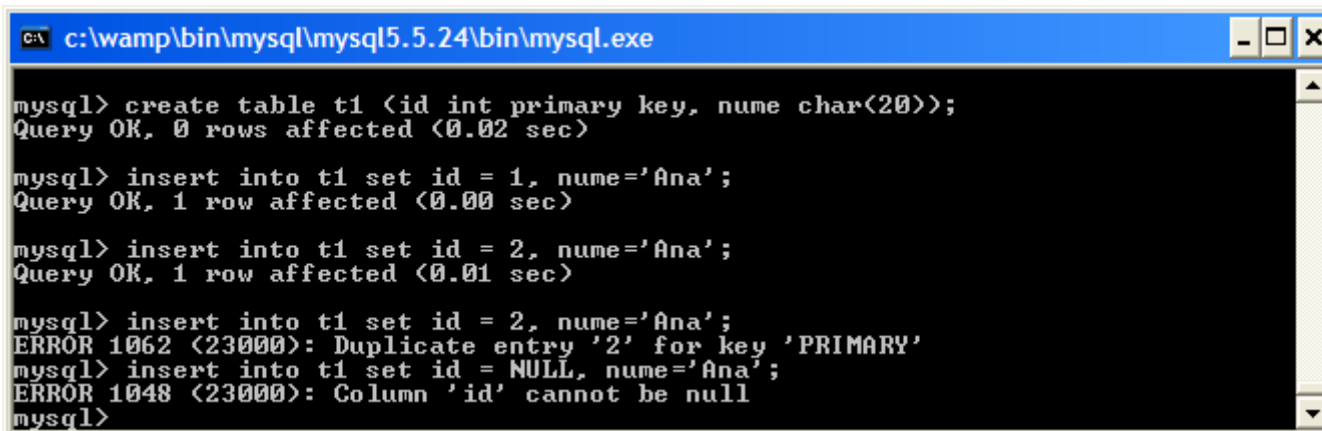
```
# Incercam sa stergem o inregistrare din tabela de adrese, a carui camp id nu este referit de nici o  
# cheie straina definita in tabela de clienti; incercarea va reusi  
DELETE FROM adresa where id = 4;
```

```
# Incercam sa stergem o inregistrare din tabela de adrese, a carui camp id este referit de o  
# cheie straina definita in tabela de clienti; incercarea va esua  
DELETE FROM adresa where id = 2;
```



```
c:\wamp\bin\mysql\mysql5.5.24\bin\mysql.exe  
mysql> DELETE FROM adresa where id = 4;  
Query OK, 1 row affected (0.03 sec)  
  
mysql> DELETE FROM adresa where id = 2;  
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails ('preda`.`client`, CONSTRAINT `client_ibfk_1` FOREIGN KEY (`adresa_id`) REFERENCES `adresa` (`id`))  
mysql>
```

## Integritatea entitatii



```
c:\wamp\bin\mysql\mysql5.5.24\bin\mysql.exe  
mysql> create table t1 (id int primary key, nume char(20));  
Query OK, 0 rows affected (0.02 sec)  
  
mysql> insert into t1 set id = 1, nume='Ana';  
Query OK, 1 row affected (0.00 sec)  
  
mysql> insert into t1 set id = 2, nume='Ana';  
Query OK, 1 row affected (0.01 sec)  
  
mysql> insert into t1 set id = 2, nume='Ana';  
ERROR 1062 (23000): Duplicate entry '2' for key 'PRIMARY'  
mysql> insert into t1 set id = NULL, nume='Ana';  
ERROR 1048 (23000): Column 'id' cannot be null  
mysql>
```

- Nu se permite:
- Existenta unei valori duplicate pentru primary key;
  - Existenta unei valori NULL pentru primary key;