

Aplicații integrate pentru întreprinderi

Laborator 2

12.10.2010



Utilizarea sistemului de gestiune pentru baze de date MySQL

Scopul laboratorului îl reprezintă folosirea sistemului de gestiune pentru baze de date MySQL în vederea creării de tabele dependente în cadrul unei baze de date precum și popularea lor cu informații care să respecte constrângerile specificate la proiectarea bazei de date.

În principiu, în cadrul acestui laborator veți recapitula principalele noțiuni însușite în cadrul cursului *Baze de Date I* din semestrul trecut.

1. Instalare

- 1.1. Sisteme de operare Linux
- 1.2. Sisteme de operare Windows

2. Utilizare: conectare și deconectare la baza de date

3. Tipuri de date

4. Proiectarea tabelor și operații de manipulare în bazele de date

5. Interogări

6. Folosirea de script-uri specifice

1. Instalare

MySQL reprezintă cea mai populară soluție *open-source* la nivel mondial pentru gestiunea bazelor de date. În cadrul laboratorului vom lucra cu MySQL Community Server¹, disponibil la <http://mysql.com/downloads/mysql/> atât pentru sisteme de operare Linux, cât și pentru sisteme de operare Windows.

1.1 Sisteme de operare Linux

Pentru sistemele de operare Linux sunt puse la dispoziție pachete .rpm care în cazul distribuțiilor care suportă comanda rpm²

```
> rpm -install [options] filename.rpm
```

se instalează ca atare, iar pentru distribuții de tipul Debian sau Ubuntu care necesită fișiere de tip .deb, este necesară mai înainte convertirea între cele două tipuri de fișiere (în acest sens, poate fi folosit un utilitar de tip Alien³):

```
alien -scripts filename.rpm  
dkpg -i filename.deb
```

¹ Pentru sistemele de operare Linux, trebuie instalat separat serverul și clientul, în timp ce pentru sistemele de operare Windows, serverul și clientul sunt incluse în același pachet.

² O listă completă pentru opțiunile suportate de comanda rpm poate fi consultată la http://linux.about.com/od/commands/l/blcmdl8_rpm.htm

³ În caz că un astfel de utilitar nu este încă instalat, acest lucru poate fi realizat prin comanda `apt-get install alien`

În cazul că se obțin erori la convertirea între cele două fișiere, programele pot fi instalate astfel:

```
# apt-get install mysql-server
# apt-get install mysql-client
```

Verificarea versiunii de mysql instalată prin intermediul acestor comenzi poate fi verificată prin comanda

```
andrei@ubuntu:~$ mysql --version
mysql Ver 14.14 Distrib 5.1.41, for debian-linux-gnu (i486) using
readline 6.1
```

1.2 Sisteme de operare Windows

Pentru sistemele de operare Windows sunt puse la dispoziție atât installere .msi, cât și arhive .zip.

Instalarea oferă posibilitatea configurării serverului, specificându-se destinația aplicației (client, server [dedicat]), utilizarea bazei de date (tranzacțională sau nu, pentru scopuri multiple), numărul de conexiuni concurente, permiterea conectării la baza de date prin conexiuni TCP/IP (dacă este necesară accesarea bazei de date din Internet, în afara calculatorului local), instalarea serverului ca serviciu Windows pornit automat, specificarea politicilor de securitate prin specificare de utilizatori cu parole asociate.

2. Utilizare: conectare și deconectare la baza de date

Conectarea la un server de baze de date se face prin comanda `mysql`, având sintaxa:

```
mysql -h host -u username -p
```

unde

`host` – reprezintă mașina pe care rulează serverul de MySQL (poate fi pornit prin comanda `mysqld`, în caz că nu a fost instalat ca serviciu care să pornească automat); nu este necesară specificarea acestui parametru în condițiile în care serverul rulează pe aceeași mașină cu clientul;

`username` – reprezintă numele utilizatorului care se conectează la server;

În continuare, va fi solicitată parola asociată utilizatorului și, în caz că aceasta este corectă, va apărea prompt-ul `mysql>` la care pot fi realizate interogări ale bazei de date.

Deconectarea de la serverul MySQL se face prin intermediul comenzii `quit`:

```
C:\Users\Andrei>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.51-community MySQL Community Server (GPL)
```

```

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> quit
Bye

C:\Users\Andrei>

```

3. Tipuri de date

În MySQL pot fi folosite trei tipuri de date:

- numerice;
- dată calendaristică.
- șiruri de caractere;

Tipurile de date numerice sunt redate în tabelul de mai jos:

BIT[(M)]
TINYINT[(M)] [UNSIGNED] [ZEROFILL]
BOOL, BOOLEAN
SMALLINT[(M)] [UNSIGNED] [ZEROFILL]
MEDIUMINT[(M)] [UNSIGNED] [ZEROFILL]
INT[(M)] [UNSIGNED] [ZEROFILL]
BIGINT[(M)] [UNSIGNED] [ZEROFILL]
FLOAT[(M,D)] [UNSIGNED] [ZEROFILL]
DOUBLE[(M,D)] [UNSIGNED] [ZEROFILL]
DOUBLE PRECISION[(M,D)] [UNSIGNED] [ZEROFILL]
REAL[(M,D)] [UNSIGNED] [ZEROFILL]
FLOAT(p) [UNSIGNED] [ZEROFILL]
DECIMAL[(M[,D])] [UNSIGNED] [ZEROFILL]
DEC[(M[,D])] [UNSIGNED] [ZEROFILL]
NUMERIC[(M[,D])] [UNSIGNED] [ZEROFILL]
FIXED[(M[,D])] [UNSIGNED] [ZEROFILL]

Observație.

SERIAL este un alias pentru BIGINT UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE.

În tabelul de mai sus, M reprezintă numărul total de cifre pe care poate fi reprezentată o valoare de tipul numeric specificat, iar D numărul de cifre care urmează după virgulă în cazul numerelor reale.

În cazul în care se specifică valoarea ZEROFILL în cazul unor coloane conținând valori numerice, acestora li se adaugă automat valoarea UNSIGNED.

Informațiile de tip dată calendaristică pot avea următoarele tipuri:

DATE
DATETIME
TIMESTAMP
TIME
YEAR[(2 4)]

Tipurile de date de tip șir de caractere sunt redade în tabelul de mai jos:

```
[NATIONAL] CHAR(M) [CHARACTER SET charset_name]  
    [COLLATE collation_name]  
[NATIONAL] VARCHAR(M) [CHARACTER SET charset_name]  
    [COLLATE collation_name]  
BINARY(M)  
VARBINARY(M)  
TINYBLOB  
TINYTEXT [CHARACTER SET charset_name] [COLLATE collation_name]  
BLOB(M)  
TEXT(M) [CHARACTER SET charset_name] [COLLATE collation_name]  
MEDIUMBLOB  
MEDIUMTEXT [CHARACTER SET charset_name] [COLLATE collation_name]  
LONGBLOB  
LONGTEXT [CHARACTER SET charset_name] [COLLATE collation_name]  
ENUM('value1','value2',...) [CHARACTER SET charset_name]  
    [COLLATE collation_name]  
SET('value1','value2',...) [CHARACTER SET charset_name]  
    [COLLATE collation_name]
```

4. Proiectarea tabelor și operații de manipulare în bazele de date

În proiectarea tabelor trebuie să se aibă în vedere câteva reguli: structura tabelului trebuie să fie normalizată, adică să respecte o formă normală, astfel încât să fie evitată apariția de anomalii la operațiile de manipulare cu baza de date; pentru fiecare tabelă în parte trebuie să se definească o cheie primară care să fie unică și care să identifice o înregistrare din baza de date; tipul de date trebuie ales astfel încât să răspundă necesităților, evitând irosirea de spațiu pe discul fizic, dar evitând situația în care anumite informații nu pot fi reținute datorită unei alocări insuficiente a unui tip de date.

Se pot vizualiza bazele de date existente prin intermediul comenzii:

```
SHOW DATABASES;
```

În cadrul unei baze de date pot fi realizate operații imediat ce e selectată:

```
USE databasename;
```

Acordarea de drepturi pentru un utilizator în cadrul unei tabele dintr-o bază de date se face prin comanda:

```
GRANT ALL ON databasename.* TO 'username'@'host';
```

Tabelele din cadrul unei baze de date (care a fost selectată anterior) se pot vizualiza prin comanda:

```
SHOW TABLES;
```

Informații suplimentare despre structura unei tabele pot fi aflate astfel:

```
DESCRIBE tablename;
```

O tabelă în baza de date este creată prin CREATE TABLE:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
  (create_definition,...)
  [table_options]
  [partition_options]
```

un exemplu de folosire fiind descris mai jos:

```
mysql> CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20),
-> species VARCHAR(20), sex CHAR(1), birth DATE, death DATE);
```

Ștergerea tabelii (ca de altfel și a bazei de date) se face prin comanda DROP:

```
DROP TABLE tablename;
DROP DATABASE databasename;
```

Adăugarea de informații poate fi realizată prin comanda INSERT sau prin comanda LOAD DATA (în acest caz sunt introduse date direct din fișier, specificându-se formatul datelor din fișier):

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
  [INTO] tbl_name [(col_name,...)]
  {VALUES | VALUE} ({expr | DEFAULT},...),(...),...
  [ ON DUPLICATE KEY UPDATE
    col_name=expr
    [, col_name=expr] ... ]
```

```
LOAD DATA [LOW_PRIORITY | CONCURRENT] [LOCAL] INFILE 'file_name'
  [REPLACE | IGNORE]
  INTO TABLE tbl_name
  [CHARACTER SET charset_name]
  [{FIELDS | COLUMNS}
    [TERMINATED BY 'string']
    [[OPTIONALLY] ENCLOSED BY 'char']
    [ESCAPED BY 'char']
  ]
  [LINES
    [STARTING BY 'string']
    [TERMINATED BY 'string']
  ]
  [IGNORE number LINES]
  [(col_name_or_user_var,...)]
  [SET col_name = expr,...]
```

Exemple de populare a tabelii creată anterior prin comanda CREATE TABLE ar putea fi:

```
mysql> INSERT INTO pet
-> VALUES ('Puffball','Diane','hamster','f','1999-03-30',NULL);
```

```
mysql> LOAD DATA LOCAL INFILE '/path/pet.txt' INTO TABLE pet
-> LINES TERMINATED BY '\r\n';
```

5. Interogări

Interogarea bazei de date se face prin comanda `SELECT`, ce are sintaxa:

```
SELECT what_to_select
FROM which_table
WHERE conditions_to_satisfy
[ORDER BY] property
[GROUP BY] criteria;
```

O interogare poate fi făcută pe mai multe tabele din baza de date între care există o relație (operație de join):

```
mysql> SELECT pet.name,
-> (YEAR(date)-YEAR(birth)) - (RIGHT(date,5)<RIGHT(birth,5)) AS age,
-> remark
-> FROM pet INNER JOIN event
-> ON pet.name = event.name
-> WHERE event.type = 'litter';
```

Se poate identifica numărul de linii dintr-un fișier care corespund unui anumit criteriu:

```
SELECT owner, COUNT(*) FROM pet GROUP BY owner;
```

Prin intermediul directivei `LIKE` adăugată la comanda `SELECT`, pot fi folosite șabloane în scopul identificării unor șiruri de caractere care se potrivesc cu modele diverse, sintaxa fiind asemănătoare ca în cazul expresiilor regulate.

6. Folosirea de script-uri specifice

Câteodată este util, în special în caz de testare, ca toate comenzile să fie grupate în cadrul unui fișier și executate secvențial.

Se poate folosi fie operatorul de indirectare sau se pot folosi comenzile `source` sau `.\`:

```
shell> mysql -h host -u user -p < filename
Enter password: *****
```

```
mysql> source filename;
mysql> .\ filename
```

În cadrul acestui laborator va trebui să proiectați bazele de date aferente temei de casă 1, să le creați, să le populați și să le interogați prin intermediul unor scripturi specifice.