

MySQL でのデータベースアカウント管理

実習用アカウント管理と、その簡略化を目指して

技術センター 情報部門コンテンツ技術班

寸田 祐樹

1. はじめに

工学部第二類3年次開講の講義「データベース」では実習で MySQL を使用する。その実習で使用するデータベースアカウントの管理を、担当されていた助教の先生の異動により、2008年度以降は私が行なうことになった。そこで、今回私が実際に行った手順と簡略化を目指した工夫を紹介する。

2. 状況の概略

実習は工学研究科情報基盤支援センターの工学部教育用コンピュータ室で行われ、MySQL サーバも工学研究科情報基盤支援センター側で用意されたものを使用する。アカウントの管理は MySQL サーバにログインして行なう。

MySQL サーバとコンピュータ室の PC には、それぞれ以下のものがインストールされている。

- ・ MySQL-server-4.1.22-0.glibc23.i386.rpm
- ・ MySQL-client-4.1.22-0.glibc23.i386.rpm

引き継いだ時点での MySQL サーバのデータベースアカウントは前年度のままで、前年度の受講生のものがそのまま残っていた。受講生のユーザ名は学生番号とし、受講生用のデータベース名は d+ ユーザ名としてある。また、一人の受講生に対して以下の2つのアカウントを用意している。

- ・ ユーザ名 @ localhost
- ・ ユーザ名 @ 接続元ドメイン名

それぞれのアカウントには自身が使用するデータベースに以下の権限が与えてあった。

```
「SELECT, INSERT, UPDATE, DELETE,  
CREATE, DROP」
```

MySQL では、ユーザ名と接続元をセットにしてアカウントとみなす。接続元には localhost やドメイン名以外にもホスト名や IP アドレスが指定できる。2つのアカウントの内、localhost は ssh など MySQL サーバに既にログインしている場合に使用し、ドメインの方のアカウントはコンピュータ室の PC から直接 MySQL サーバにアクセスする場合に使用する。受講生が混乱しないよう2つのアカウントのパスワードは統一しておく必要がある。

3. データベースアカウントの管理

(1) Windows からの MySQL 接続

データベースアカウント管理のために、筆者の研究室居室 Windows マシン (Windows XP) からの MySQL 接続環境を構築する。

MySQL 公式サイト¹⁾ からインストーラなしの mysql-noinstall-5.1.31-win32.zip をダウンロードし、適当な場所へ展開する。展開して出来た mysql-5.1.31-win32 フォルダを C ドライブ直下に置く。(実際にはデスクトップのような日本語ディレクトリの下層でなければ、どこにおいても良い。ここではディレクトリの移動の手間を省くため、C ドライブ直下とした。パーティションを切っただけならば、D ドライブ等でも良い。)

(2) MySQL サーバへの接続

コマンドプロンプトを起動し、以下のようにカレントディレクトリを変更する。

```
> cd %mysql-5.1.31-win32%\bin
```

管理者権限を持つユーザでログインする。

> mysql -h 接続先ホスト名 -u ユーザ名 -p
パスワードを入力して接続が完了する。
また、切断する場合は quit (¥q) と入力する。
mysql > quit

(3) アカウントの削除手順

必要のないアカウント（前年度の受講者）の削除を行う前に、登録されているユーザを確認する。

```
mysql > SELECT host, user FROM  
mysql.user;  
また、データベースの一覧も確認しておく。  
mysql > SHOW DATABASES;
```

ここから、学生番号 B001234 という架空の受講生を例に、アカウントの削除手順を記す。説明のため、ここからは接続元ドメイン名を xyz.hiroshima-u.ac.jp として記述する。

ユーザ権限の削除

REVOKE 構文により、B001234@localhost がデータベース dB001234 に対して持っている全ての権限を取り去る。

```
mysql > REVOKE ALL ON dB001234.*  
FROM B001234@'localhost';  
同様に、ドメインの方のアカウントからも全ての権限を取り去る  
mysql > REVOKE ALL ON dB001234.*  
FROM B001234@'%.xyz.hiroshima-u.ac.jp';
```

アカウントの削除

DROP USER 構文により、B001234@localhost を削除する。
mysql > DROP USER B001234@localhost;
同様に、ドメインの方のアカウントも削除する。
mysql > DROP USER
B001234@'%.xyz.hiroshima-u.ac.jp';

データベースの削除

DROP DATABASE 構文により、データベース dB001234 を削除する。

```
mysql > DROP DATABASE dB001234;
```

権限テーブルの再読み込み

権限テーブルの再読み込みを行ない、これまで行った変更を反映させる。

```
mysql > FLUSH PRIVILEGES;
```

(4) アカウントの作成手順

学生番号 B001234 という架空の受講生を例に、localhost アカウントの作成手順を記す。なお、実際にはパスワードは8文字のランダムな英数字に設定するが、便宜上ここでは password という文字列を設定することにする。

データベースの作成

CREATE DATABASE 構文により、B001234 が使用するデータベースを作成する。

```
mysql > CREATE DATABASE dB001234;
```

アカウントの作成

GRANT 構文により、作成したデータベース dB001234 に対して以下の権限を持つアカウントを作成する。

「SELECT, INSERT, UPDATE, DELETE, CREATE, DROP」

```
mysql > GRANT SELECT, INSERT,  
UPDATE, DELETE, CREATE, DROP ON  
dB001234.* TO B001234@'localhost'  
IDENTIFIED BY 'password';  
同様に、ドメインの方のアカウントも作成する。  
mysql > GRANT SELECT, INSERT,  
UPDATE, DELETE, CREATE, DROP ON  
dB001234.* TO  
B001234@'%.xyz.hiroshima-u.ac.jp'  
IDENTIFIED BY 'password';
```

権限テーブルの再読み込み

権限テーブルの再読み込みを行ない、これまで行った変更を反映させる。

```
mysql > FLUSH PRIVILEGES;
```

4. 管理上の問題点

2008年度の受講生は36名であり、TA2名を合わせると38名である。

1名のアカウントを削除するのに、前述のようにクエリを6つ要する。これでは、時間が掛かるとともにミスを誘発しやすい。そのため、アカウントの作成と削除の操作を簡略化するための工夫を行なうこととした。

5. 簡略化の工夫

(1) MySQL をバッチモードで使用

MySQL はバッチモードでも実行できる。実行するクエリを全て外部ファイルに記述し、リダイレクトで渡せば1文1文クエリを入力して実行しなくて済む。

(2) バッチファイルをスクリプトで生成

各ユーザでデータベース名、ユーザ名、パスワードは異なるが、それ以外のコマンドは共通である。そのため、専用のスクリプトを作成すれば、バッチモードで使用する外部ファイルを手動で作成しなくても良くなると思い、スクリプトの作成に取り掛かることにした。

6. スクリプトの作成と実行

データベース名、ユーザ名、パスワードを変数とし、必要なクエリをユーザ分繰り返し出力するスクリプトを作成する。

(1) 仕様

- ・スクリプト言語は Perl を使用する。
- ・データ入力は CSV ファイルを使用する。
- ・スクリプトはアカウント作成用と削除用の2種類作成する。

(2) Perl の実行環境

筆者は以前より研究室居室の Windows マシンに Cygwin 上での Perl 実行環境を構築しており、これをそのまま使用した。

Cygwin+Perl の導入は概ね以下の通りである。Cygwin 公式サイト³⁾ から setup.exe をダウンロードし、実行する。インストーラが起動するので適宜選択し、パッケージ選択画面で Interpreters から Perl を選択する。

(3) 入力用 CSV ファイル

「ユーザ名」と「パスワード」をカンマで区切った CSV 形式のファイルを入力として用意する。必要な情報は以下の2つ。

- ・ユーザ名：学生番号
- ・パスワード：8文字のランダムな英数字

例 . list.csv

```
B001000,XnUcl1w9
B001001,NNXI4MZ9
B001002,uXVdf9fJ
B001003,suQCoZP6
B001004,ofRJHHY0
B001005,VzulJnw6
B001006,UxA6fAUQ
B001007,uKoiYubs
B001008,iANbXl0a
B001009,Qwz2TKMT
B001010,zzBcPZN0
```

(4) スクリプトファイル

以下の2つのスクリプトファイルを作成した。FLUSH PRIVILEGES; の1文は最後に1回だけ行なえば良いので、ループの外とした。

```
アカウント作成用スクリプト mkacfile.pl
#!/usr/bin/perl
open(FILE,"<$ARGV[0]>");
while($line=<FILE>)
```

```

{
chomp($line);
@data=split(/,/, $line);
print "CREATE DATABASE d$data[0];\n";
print "GRANT SELECT,INSERT,UPDATE,
DELETE,CREATE,DROP ON d$data[0].* TO
$data[0]\%@'localhost'
IDENTIFIED BY '$data[1]';\n";
print "GRANT SELECT,INSERT,UPDATE,
DELETE,CREATE,DROP ON d$data[0].* TO
$data[0]\%@'%xyz.hiroshima-u.ac.jp'
IDENTIFIED BY '$data[1]';\n";
}
close(FILE);
print "FLUSH PRIVILEGES;\n";

```

アカウント削除用スクリプト delacfile.pl

```

#!/usr/bin/perl
open(FILE,"<$ARGV[0]>");
while($line=<FILE>)
{
chomp($line);
@data=split(/,/, $line);
print "REVOKE ALL ON d$data[0].* FROM
$data[0]\%@'localhost';\n";
print "REVOKE ALL ON d$data[0].* FROM
$data[0]\%@'%xyz.hiroshima-u.ac.jp';\n";
print "DROP USER
$data[0]\%@'localhost';\n";
print "DROP USER
$data[0]\%@'%xyz.hiroshima-u.ac.jp';\n";
print "DROP DATABASE d$data[0];\n";
}
close(FILE);
print "FLUSH PRIVILEGES;\n";

```

(5) スクリプトによるバッチファイル生成

作成したスクリプトを使用し、アカウント作成と削除のためのバッチファイルを生成する。

データ入力用に CSV ファイルを用意する。ただし、Perl の chomp 関数が正しく動作するように改行コードを LF として保存しておく。

アカウント作成バッチファイル生成

```
$ ./mkacfile.pl list.csv > mkac.bat
```

アカウント削除バッチファイル生成

```
$ ./delacfile.pl list.csv > delac.bat
```

7. バッチファイルでのアカウント管理

(1) Cygwin 上での MySQL 接続

Cygwin 上で MySQL を使用できるようにする。C ドライブにある mysql-5.1.31-win32 のフォルダ全体をエクスプローラ等を使用して、Cygwin をインストールしたフォルダの下の /usr/local/ に移動する。

また、mysql にパスを通しておく。Cygwin 上のホームディレクトリにある .bashrc に以下の記述を追加する。

```
export PATH=$PATH:/usr/local/mysql-5.1.30-
win32/bin
```

すぐ作業に移る場合は、以下のようにすると変更が反映される。

```
$ source .bashrc
```

(2) アカウントの作成

```
$ mysql -h 接続先ホスト名 -u ユーザ名 -p <
mkac.bat
```

(3) アカウントの削除

```
$ mysql -h 接続先ホスト名 -u ユーザ名 -p <
delac.bat
```

以上で、CSV ファイルは用意する必要があるものの、一応アカウント管理を簡略化できた。

8. パスワードの自動生成

ここまででは、パスワードは別途パスワード

生成ツールによって生成し、データ入力用の CSV ファイルを手動で作成していた。せっかくなので、これを何とか自動化したいと思う。

Perl のモジュールにはランダムな文字列を生成する `String::Random` がある。これを使用し、入力用の `list.csv` を生成するスクリプトを作成する。

(1) `String::Random` モジュールのインストール

CPAN を使ってインストールする。以下のように CPAN を起動する。

```
$ perl -MCPAN -e shell
```

初回はいくつか聞かれるがデフォルトのままが良いので、Enter を押す。

起動すると、以下のようになる。

```
cpan[1] >
```

以下のように、モジュールをインストールする。

```
cpan[1] > install String::Random
```

また何か聞かれたら、Enter で良い。

インストールが完了したら、CPAN を終了する。

```
cpan[2] > q
```

(2) 生成する `list.csv` について

パスワードは、アルファベット大文字小文字と数字からなる8桁のものとする。ただし、大文字のオーは数字のゼロと区別し難いので、大文字のオーは使用しない。

また、アカウントを配布するときのことも考え、「氏名」の情報も追加する。既に作成した2つのスクリプトには変更を加えないで済むよう、`list.csv` の書式は以下のようにする。

ユーザ名、パスワード、氏名

(3) `list.csv` 作成スクリプト `mklist.pl`

```
#!/usr/bin/perl
open(FILE,"<$ARGV[0]>");
<FILE>;
while($line=<FILE>)
{
```

```
    chomp($line);
    @data=split(/,/, $line);
    use String::Random;
    $password=new String::Random->randregex
    ('[a-zA-NP-Z0-9]{8}');
    print "$data[8],$password,$data[9]\n";
}
close(FILE);
```

(4) `list.csv` の生成

学生番号のデータは「もみじ」から出力されたもの (CSV ファイル) を使用する。そのままでは余分な情報 (ダブルクォーテーション) が含まれているので、一度 Excel で読み込み、保存し直す。必要なら TA の情報も追加しておく。
\$./delacfile.pl もみじから出力されたファイル
> list.csv

以上で、もみじから出力されたファイルからパスワードを自動生成した `list.csv` を生成できた。

9. 終わりに

初めは MySQL にほとんど触れたことがないところから出発したが、これまでの業務で培った知識を用いて、アカウント管理の簡略化を行なった。来年度以降は作成した3つのスクリプトを使用することで、アカウント管理の手間を軽減できると思う。

10. 参考 URL

1) MySQL 公式サイト

<http://www.mysql.com/>

2) MySQL 4.1 リファレンスマニュアル

<http://dev.mysql.com/doc/refman/4.1/ja/>

3) Cygwin 公式サイト

<http://www.cygwin.com/>

4) CPAN `String::Random` のページ

<http://search.cpan.org/~steve/String-Random-0.22/lib/String/Random.pm>