

哈工大计算机学院系统安全

实验三

数据库用户管理的权限管理设计和实现

学号：1110320116

姓名：李明哲

指导老师：张玥

时间：2013 年 11 月 24 日

一. 实验内容

在熟练掌握 MySQL 基本权限管理命令、SQL 语言以及学习简单数据库系统的设计、数据库用户权限设定的实现，包括特定场景下地数据表创建和管理、数据库用户的创建和合理的权限分配，权限分配细化到数据库、表和列。

二. 实验设计

1. 【环境搭建】

MySQL 有很多版本，主要有 MySQL Sever5.5，wamp 集成软件等。通过测试选用 MySQL sever5.5（安装时会同时安装 MySQL workbench，后者是前者的图形化界面）。

2. 【数据库设计】

考虑实际使用场景，建立名为 lmz 的数据库来模拟学生老师课程信息管理。建立五个表：

- (1) 学生的自然信息，如学号、姓名、性别、出生日期、所在班级、所在年级、入学时间、家庭住址等。
- (2) 教师的自然信息，如教师工号、姓名、性别、出生日期、职称、所在院系、家庭住址等。
- (3) 课程的自然信息，如课程编号、课程名称、课程类别（必修、限选、任选）、课程的学分、开课院系、备注。
- (4) 老师-课程联系表，包含课程号、教师工号、课程学期
- (5) 学生-课程联系表，包含课程号、选课学生学号、学生得分、备注。

3. 【权限设计】

考虑数据库敏感信息的机密性，对权限设计如下：

学生对学生表中学生姓名、学号有查询权限，对其他信息不应能随便查询。考虑其他信息的不可变性，学生仅对地址有修改权限。对老师信息仅有查看姓名的权限，而不可以查看其他信息造成老师信息泄露。作为校学生，对所有课程信息有查询权限。对于教师课程联系表，和学生课程联系表，均应有查看权限，方便联系授课老师或者比较同课程同学间的成绩。

老师身份可以对学生的姓名。学号进行查询，因为学生个人信息设计敏感内容，且不需要被老师知晓；老师可以查看教师表的所有信息，并对个人的地址进行修改；课程表可以被教师身份查询，由于老师-课程联系表设计的内容是学分、授课老师、课程类别、开课院系，这些信息不应该可以被老师随意修改，因此教师对该表仅有 `select` 权限；学生-课程联系表，老师除了必须应有的查询所有信息之外，还应对自己课程拥有更改成绩的权限。

而系统管理员 `root` 身份，则应成为唯一能对数据库所有表进行查询、修改、插入权限的人。只有这样，才能正常管理数据库信息，应对所有情况。如下，将身份和对应各个表细化到列的权限写入如下的表中。

	student	teacher	S_c	T_c	course
root	all	all	all	all	all
Student_test	Select sname,sid Update saddr	Select tname	Select all	Select all	Select all
Teacher_test	Select sname, sid	Select all Update taddr	Select all Update score,note	Select all	Select all

4. 【函数封装】

根据实验要求，结合实际情况，当数据库真正使用时，应对常用的操作进行封装成函数，便于批量处理。因此，我选择将对于不同身份进行权限授予的过程封装成为函数。权限授予的内容与上表一致。

三. 实验过程

1. 建立数据库，名为 'lmz'。
2. 建立 student, teacher, course, s_c, t_c 五个表。

```
MySQL 5.5 Command Line Client

mysql> create database lmz;
Query OK, 1 row affected (0.01 sec)

mysql> use lmz;
Database changed
mysql> show tables;
Empty set (0.00 sec)

mysql> CREATE TABLE `lmz`.`course` (
  -> `c_id` INT NOT NULL ,
  -> `c_name` VARCHAR(255) NULL ,
  -> `c_category` ENUM('必修','限选','任选') NULL ,
  -> `c_credit` INT NULL ,
  -> `c_department` VARCHAR(255) NULL ,
  -> `c_note` VARCHAR(255) NULL ,
  -> PRIMARY KEY (`c_id`) );
Query OK, 0 rows affected (0.10 sec)

mysql> CREATE TABLE `lmz`.`student` (
  -> `s_id` INT NOT NULL , `s_name` VARCHAR(45) NULL ,
  -> `s_gender` ENUM('female','male') NULL DEFAULT 'male' ,
  -> `s_birthday` DATE NULL ,
  -> `s_class` INT NULL ,
  -> `s_department` VARCHAR(255) NULL ,
  -> `s_admission_date` DATE NULL ,
  -> `s_addr` VARCHAR(255) NULL ,
  -> PRIMARY KEY (`s_id`) );
Query OK, 0 rows affected (0.08 sec)

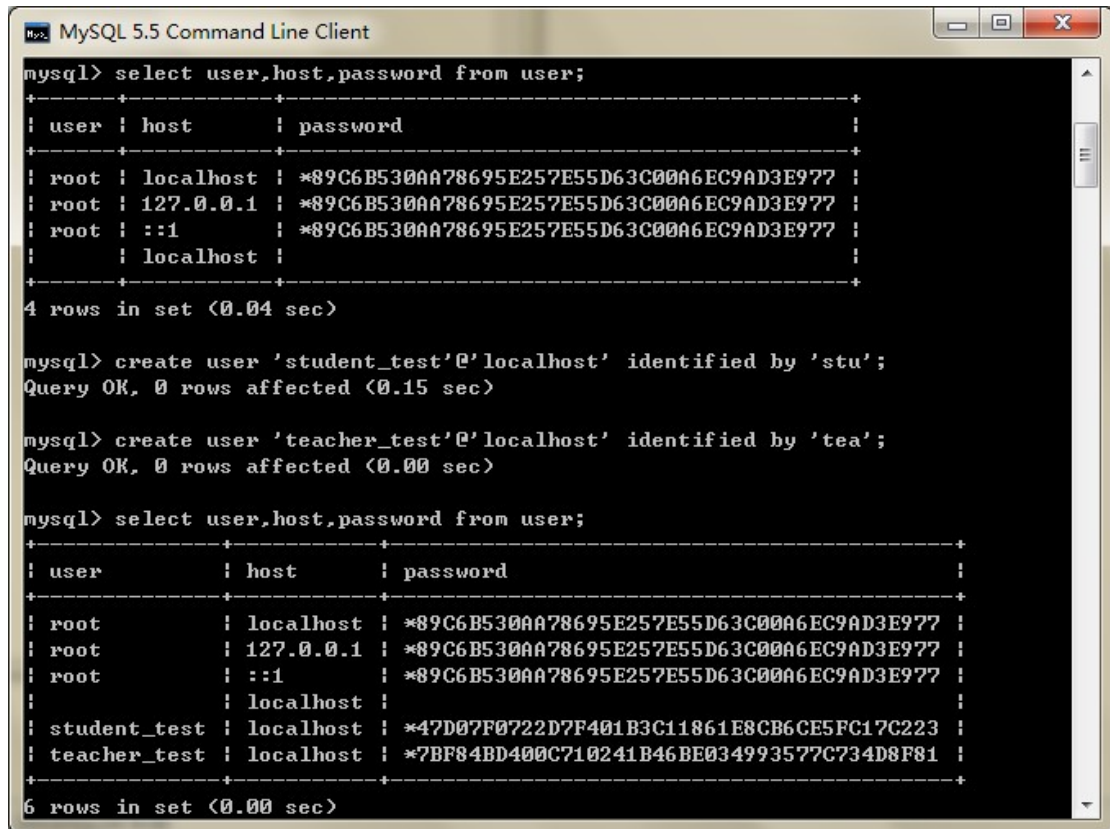
mysql> CREATE TABLE `lmz`.`teacher` (
  -> `t_id` INT NOT NULL ,
  -> `t_name` VARCHAR(255) NULL ,
  -> `t_gender` ENUM('female','male') NULL DEFAULT 'male' , `t_birthday` DA
TE NULL ,
  -> `t_title` VARCHAR(255) NULL ,
  -> `t_department` VARCHAR(255) NULL ,
  -> `t_addr` VARCHAR(255) NULL ,
  -> PRIMARY KEY (`t_id`) );
Query OK, 0 rows affected (0.09 sec)

mysql> CREATE TABLE `lmz`.`t_c` (
  -> `tc_cid` INT NOT NULL ,
  -> `tc_tid` INT NOT NULL ,
  -> `tc_term` ENUM('春季','秋季') NULL ,
  -> PRIMARY KEY (`tc_cid`, `tc_tid`) );
Query OK, 0 rows affected (0.12 sec)

mysql>
mysql> CREATE TABLE `lmz`.`s_c` (
  -> `sc_cid` INT NOT NULL ,
  -> `sc_sid` INT NOT NULL ,
  -> `sc_score` INT NULL ,
  -> `sc_note` VARCHAR(255) NULL ,
  -> PRIMARY KEY (`sc_cid`, `sc_sid`) );
Query OK, 0 rows affected (0.15 sec)
```

3. 建立名为 student_test 和 teacher_test 的两个用户以代表学生身份和老师身份。通过查看 user, host, passwd from user, 查看是

否成功。



```
mysql> select user,host,password from user;
+-----+-----+-----+
| user | host      | password |
+-----+-----+-----+
| root | localhost | *89C6B530AA78695E257E55D63C00A6EC9AD3E977 |
| root | 127.0.0.1 | *89C6B530AA78695E257E55D63C00A6EC9AD3E977 |
| root | ::1       | *89C6B530AA78695E257E55D63C00A6EC9AD3E977 |
|      | localhost |          |
+-----+-----+-----+
4 rows in set (0.04 sec)

mysql> create user 'student_test'@'localhost' identified by 'stu';
Query OK, 0 rows affected (0.15 sec)

mysql> create user 'teacher_test'@'localhost' identified by 'tea';
Query OK, 0 rows affected (0.00 sec)

mysql> select user,host,password from user;
+-----+-----+-----+
| user      | host      | password |
+-----+-----+-----+
| root      | localhost | *89C6B530AA78695E257E55D63C00A6EC9AD3E977 |
| root      | 127.0.0.1 | *89C6B530AA78695E257E55D63C00A6EC9AD3E977 |
| root      | ::1       | *89C6B530AA78695E257E55D63C00A6EC9AD3E977 |
|          | localhost |          |
| student_test | localhost | *47D07F0722D7F401B3C11861E8CB6CE5FC17C223 |
| teacher_test | localhost | *7BF84BD400C710241B46BE034993577C734D8F81 |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

4. 分别使用 student_test 和 teacher_test 进入 mysql, use lmz 数据库。发现均无权限使用。

```
C:\windows\system32\cmd.exe - mysql -u teacher_test -p
sword: YES>

C:\Users\Mlitz>mysql -u student_test -p
Enter password: ***
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 5.5.25 MySQL Community Server (GPL)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> use lmz;
ERROR 1044 (42000): Access denied for user 'student_test'@'localhost' to databas
e 'lmz'
mysql> quit
Bye

C:\Users\Mlitz>mysql -u teacher_test -p
Enter password: ***
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 5.5.25 MySQL Community Server (GPL)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> use lmz;
ERROR 1044 (42000): Access denied for user 'teacher_test'@'localhost' to databas
e 'lmz'
半:
```

5. 编写封装起来的授权函数。分为 power to student--test4 和 power to teacher--test。权限内容与表一致。在这里可以看到，权限实现细化到列的目标。符合实际使用需求。

```
mysql> use lmz
Database changed
mysql> delimiter $$
mysql> create procedure test4()
-> begin
-> grant select <s_name> on student to 'student_test'@'localhost';
-> grant select <s_id> on student to 'student_test'@'localhost';
-> grant select <t_name> on teacher to 'student_test'@'localhost';
-> grant select on course to 'student_test'@'localhost';
-> grant select on s_c to 'student_test'@'localhost';
-> grant select on t_c to 'student_test'@'localhost';
-> grant update <s_addr> on student to 'student_test'@'localhost';
-> end $$
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> use lmz
Database changed
mysql> delimiter $$
mysql> create procedure test()
-> begin
-> grant select <s_name> on student to 'teacher_test'@'localhost';
-> grant select <s_id> on student to 'teacher_test'@'localhost';
-> grant select on teacher to 'teacher_test'@'localhost';
-> grant select on course to 'teacher_test'@'localhost';
-> grant select on t_c to 'teacher_test'@'localhost';
-> grant select on s_c to 'teacher_test'@'localhost';
-> grant update <sc_score> on s_c to 'teacher_test'@'localhost';
-> grant update <sc_note> on s_c to 'teacher_test'@'localhost';
-> end $$
Query OK, 0 rows affected (0.07 sec)
```

6. 函数编写完成之后，调用 call + 函数名。然后调用 show grants，查看是否成功赋权。


```
C:\windows\system32\cmd.exe - mysql -u student_test -p

C:\Users\Mlitz>mysql -u student_test -p
Enter password: ***
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 36
Server version: 5.5.25 MySQL Community Server (GPL)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> use lmz
Database changed
mysql> show grants
-> ;
+-----+
| Grants for student_test@localhost |
+-----+
| GRANT USAGE ON *.* TO 'student_test'@'localhost' IDENTIFIED BY PASSWORD '*47D07F0722D7F401B3C11861E8CB6CE5FC17C223' |
| GRANT SELECT ON 'lmz'`.`course` TO 'student_test'@'localhost' |
| GRANT SELECT (t_name) ON 'lmz'`.`teacher` TO 'student_test'@'localhost' |
| GRANT SELECT ON 'lmz'`.`t_c` TO 'student_test'@'localhost' |
| GRANT SELECT (s_class, s_name, s_id), UPDATE (s_addr) ON 'lmz'`.`student` TO 'student_test'@'localhost' |
| GRANT SELECT ON 'lmz'`.`s_c` TO 'student_test'@'localhost' |
+-----+
6 rows in set (0.00 sec)
```

7. 使用 student_test 登陆，进行基本操作验证。

```
C:\windows\system32\cmd.exe - mysql -u teacher_test -p

Your MySQL connection id is 39
Server version: 5.5.25 MySQL Community Server (GPL)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> use lmz
Database changed
mysql> update student.s_addr from student where s_id="1";
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'from
student where s_id="1"' at line 1
mysql> select * from teacher;
ERROR 1143 (42000): SELECT command denied to user 'student_test'@'localhost' for
column 't_id' in table 'teacher'
mysql> update student set s_addr="china";
Query OK, 5 rows affected (0.10 sec)
Rows matched: 5 Changed: 5 Warnings: 0

mysql> quit
```

发现未授权的部分不可执行，但是可以修改学生地址，并显示修改成功。

8. 使用 teacher_test 登陆，进行相应的操作。查看，及修改某课程成绩。

```
C:\windows\system32\cmd.exe - mysql -u teacher_test -p

Database changed
mysql> select * from course;
+-----+-----+-----+-----+-----+-----+
| c_id | c_name | c_category | c_credit | c_department | c_note |
+-----+-----+-----+-----+-----+-----+
| 20001 | os     | ???       | 3       | cs           | test   |
| 10002 | lab    | ???       | 2       | cs           | good   |
| 10003 | design | ???       | 1       | cs           | good   |
| 20004 | computer | ???     | 2       | cs           | good   |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

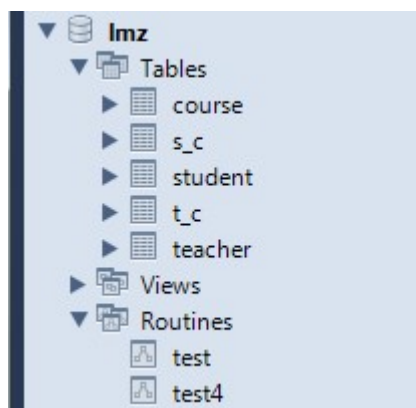
mysql> select * from s_c;
+-----+-----+-----+-----+
| sc_cid | sc_sid | sc_score | sc_note |
+-----+-----+-----+-----+
| 10001 | 1      | 90      | good    |
| 10002 | 2      | 80      | good    |
| 10003 | 3      | 85      | good    |
| 10004 | 4      | 88      | good    |
| 10005 | 5      | 89      | good    |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> update s_c set sc_score="100" where sc_cid="10001";
```

```
C:\windows\system32\cmd.exe - mysql -u teacher_test -p
mysql> select * from s_c;
+-----+-----+-----+-----+
| sc_cid | sc_sid | sc_score | sc_note |
+-----+-----+-----+-----+
| 10001  | 1      | 100      | good    |
| 10002  | 2      | 100      | good    |
| 10003  | 3      | 100      | good    |
| 10004  | 4      | 100      | good    |
| 10005  | 5      | 100      | good    |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

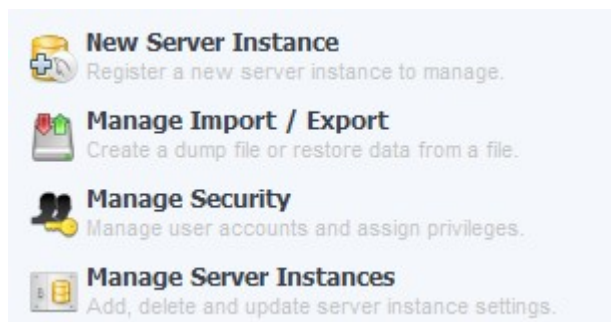
mysql> _
```

9. 主要过程完成实现。由于 workbench 是 mysql sever5.5 对应的图形界面程序，lmz 数据库在后者中可以看到并执行。并且由于实际操作中需要拥有收回权限的过程，在下面 workbench 中将有体现。

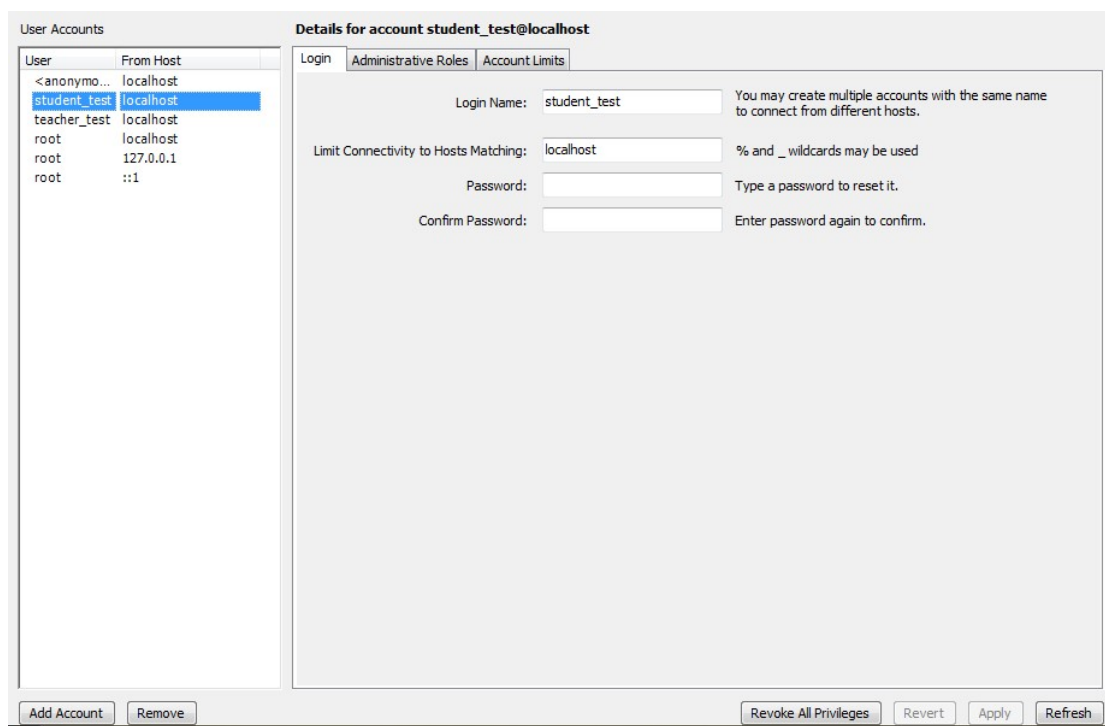


	t_id	t_name	t_gender	t_birthday	t_title	t_department	t_addr
▶	10001	ljz	male	1973-01-01	full	cs	haerbin
	10002	bxy	female	1970-01-01	associate	cs	haerbin
	10003	zhy		1975-01-01	full	cs	haerbin
	10004	ljz		1978-01-01	associate	cs	beijing

在 mysql sever 5.5 中建立的数据库 lmz。



选择 manage security 可以进行权限管理。



显示了已有的 mysql 的 user. 并对已有的权限在左下方提供权限的 revoke。体现了实验的实际意义。

四. 实验收获

1. 通过每次实验, 对数据库信息权限赋予收回有了更透彻的理解和掌握。在实际使用当中, 我们应该充分考虑到具体意义和价值, 比如数据信息中很容易包括敏感信息, 应该考虑对于不同访问数据库的角色

赋予不同的权限。事实上，单单将权限细化到表是远远不够的，因为一些表中经常同时存在敏感信息和必须要查询修改到的信息，所以掌握如何将权限细化到列显得尤为重要。

2. 老师指出，应该在权限授予的同时考虑权限收回的情况，这一点做的时候有疏忽。仔细思考，确实十分重要。因为实际中同一角色的权限可能在不同时间是不同的，必须有收回权限的过程，这样才能良好的管理数据库。

3. 当大数据库进行运营时，函数封装显得很必要。一些常用的指令封装在一起，在处理大数据的时候能够带来意想不到的效益，提高效率，减小成本。因此模拟的授权收权过程也很有价值。

4. 系统安全的三次实验，都是在实际应用中息息相关的。具有很高的实际意义和价值。以数据库权限管理设计与实现来看，我们首先需要培养系统安全包括数据库安全的基本素养，对于敏感信息要懂得保护，同时也不能耽误正常的使用。同时要尽可能的将授权进行细粒度的划分。实现权限的精确管理。在实现方面，不但要掌握数据库知识，还要领悟一些实现技巧，减小维护消耗，提高实现效率。通过这次试验，着实学到了很多。感谢老师！