# 实验 2.1 自主存取控制实验

（1）实验目的

掌握自主存取控制权限的定义和维护方法。

（2）实验内容和要求

定义用户、角色，分配权限给用户、角色，回收权限，以相应的用户名登录数据库验证权限分配是否正确。选择一个应用场景，使用自主存取控制机制设计权限分配。可以采用两种方案。方案一：采用 SYSTEM 超级用户登录数据库，完成所有权限分配工作，然后用相应用户名登陆数据库以验证权限分配正确性；方案二：采用 SYSTEM 用户登陆数据库创建三个部门经理用户，并分配相应的权限，然后分别用三个经理用户名登陆数据库，创建相应部门的 USER, ROLE，并分配相应权限。下面的实验报告示例，采用实验方案一。验证权限分配之前，请备份好数据库；针对不同用户所具有的权限，分别设计相应的 SQL 语句加以验证。

（3）实验重点和难点

实验重点：定义角色，分配权限和回收权限。

实验难点：实验方案二实现权限的再分配和回收。

**实验采用方案一进行，在 linux 下安装 MYSQL 进行本次实验。**

## 一、实验过程

**1. 使用 root 用户登录，root 用户先创建数据库 test_tbl 以供测试：**

```
mysql> CREATE DATABASE test_tbl;
Query OK, 1 row affected (0.00 sec)
```

查看 test_tbl 是否创建成功：

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| test_tbl           |
+--------------------+
5 rows in set (0.00 sec)
```

*** 创建 table students：**

```
mysql> USE test_tbl;
Database changed
mysql> SHOW TABLES;
Empty set (0.01 sec)

mysql> CREATE TABLE students (id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    -> name VARCHAR(20),
    -> department VARCHAR(30));
Query OK, 0 rows affected (0.26 sec)

mysql> SHOW TABLES;
+--------------------+
| Tables_in_test_tbl |
+--------------------+
| students           |
+--------------------+
1 row in set (0.00 sec)
```

可以看到

可以看到 students 创建成功

插入 3 条数据并查看：

```
mysql> INSERT INTO students (id, name, department) VALUES (NULL, "ugnamsung", "ES
");
Query OK, 1 row affected (1.82 sec)

mysql> INSERT INTO students (id, name, department) VALUES (NULL,"freakkid","ES");

Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO students (id, name, department) VALUES (NULL,"huangnanxuan","C
hinese");
Query OK, 1 row affected (0.03 sec)
```

```
mysql> SELECT * FROM students;
+----+--------------+------------+
| id | name         | department |
+----+--------------+------------+
|  1 | ugnamsung    | ES         |
|  2 | freakkid     | ES         |
|  3 | huangnanxuan | Chinese    |
+----+--------------+------------+
3 rows in set (0.00 sec)
```

*** 创建 table courses：**

```
mysql> CREATE TABLE courses (id INT NOT NULL PRIMARY KEY ,
    -> name VARCHAR(20),
    -> department VARCHAR(30));
Query OK, 0 rows affected (0.29 sec)
```

插入 2 条数据：

```
mysql> INSERT INTO courses (id, name, department) VALUES (1,"DBMS","ES");
Query OK, 1 row affected (0.05 sec)

mysql> INSERT INTO courses (id, name, department) VALUES (2,"HignMath","ES");
Query OK, 1 row affected (0.03 sec)

mysql> SELECT * FROM courses;
+----+----------+------------+
| id | name     | department |
+----+----------+------------+
|  1 | DBMS     | ES         |
|  2 | HignMath | ES         |
+----+----------+------------+
2 rows in set (0.00 sec)
```

总共创建两个 table：

```
mysql> SHOW TABLES;
+--------------------+
| Tables_in_test_tbl |
+--------------------+
| courses            |
| students           |
+--------------------+
2 rows in set (0.00 sec)
```

**2.创建用户并分配权限：**

*** 创建用户 'ugnamsung'@'localhost'：**

```
mysql>  create user 'ugnamsung'@'localhost' identified by 'guset*ABC*1000';
Query OK, 0 rows affected (0.00 sec)
```

查看用户权限，只有 USAGE 权限，等同于无特权：

```
mysql> show grants for "ugnamsung"@"localhost";
+-----------------------------------------------+
| Grants for ugnamsung@localhost                |
+-----------------------------------------------+
| GRANT USAGE ON *.* TO 'ugnamsung'@'localhost' |
+-----------------------------------------------+
1 row in set (0.00 sec)
```

*** 创建用户 huangnx，将对 test_tbl select 的权限分配给 huangnx：**

```
mysql> GRANT SELECT
    -> ON test_tbl.students
    -> TO 'huangnx'@'localhost'
    -> IDENTIFIED BY 'guset*ABC*123';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> show grants for "huangnx"@"localhost";
+-----------------------------------------------------------------+
| Grants for huangnx@localhost                                    |
+-----------------------------------------------------------------+
| GRANT USAGE ON *.* TO 'huangnx'@'localhost'                     |
| GRANT SELECT ON `test_tbl`.`students` TO 'huangnx'@'localhost' |
+-----------------------------------------------------------------+
2 rows in set (0.00 sec)
```

*** 创建用户 freakkid，将 GRAND_PRIV 权限分配给 freakkid：**

```
mysql>  create user 'freakkid'@'localhost' identified by 'guset*ABC*1000';
Query OK, 0 rows affected (0.00 sec)

mysql> show grants for "freakkid"@"localhost";
+-----------------------------------------------+
| Grants for freakkid@localhost                 |
+-----------------------------------------------+
| GRANT USAGE ON *.* TO 'freakkid'@'localhost' |
+-----------------------------------------------+
1 row in set (0.00 sec)
```

```
mysql> grant all privileges on *.* to 'freakkid'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> show grants for 'freakkid'@'localhost';
+-----------------------------------------------------------+
| Grants for freakkid@localhost                             |
+-----------------------------------------------------------+
| GRANT ALL PRIVILEGES ON *.* TO 'freakkid'@'localhost' |
+-----------------------------------------------------------+
1 row in set (0.00 sec)
```

*** 查看所有已创建的用户：**

```
mysql> SELECT CONCAT(QUOTE(user), "@", QUOTE(host)) UserAccount  FROM mysql.user;

+-------------------------------+
| UserAccount                   |
+-------------------------------+
| 'debian-sys-maint'@'localhost' |
| 'freakkid'@'localhost'         |
| 'huangnx'@'localhost'          |
| 'mysql.sys'@'localhost'        |
| 'root'@'localhost'             |
| 'ugnamsung'@'localhost'        |
+-------------------------------+
6 rows in set (0.00 sec)
```

可以看到三个用户创建成功。

**3. 验证权限分配正确性**

*** 登录用户 ugnamsung**

```
nanxuan@nanxuan-Lenovo-G40-70m:~$ mysql -u ugnamsung -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.7.17-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2016, 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> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
+--------------------+
1 row in set (0.00 sec)
```

用户 ugnamsung 只有 USAGE 权限，看不到 root 创建的数据库，无法对其进行任何操作。

*** 登录用户 huangnx**

```
nanxuan@nanxuan-Lenovo-G40-70m:~$ mysql -u huangnx -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.7.17-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2016, 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> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| test_tbl           |
+--------------------+
2 rows in set (0.01 sec)
```

```
mysql> SHOW tables;
+--------------------+
| Tables_in_test_tbl |
+--------------------+
| students           |
+--------------------+
1 row in set (0.00 sec)
```

用户 huangnx 可以看到 root 创建的数据库只有 test_tbl，并且也只能看到 test_tbl 的 students 一个表格，看不到 courses。

测试 huangnx 的权限：

SELECT　成功:

```
mysql> SELECT * FROM students;
+----+--------------+------------+
| id | name         | department |
+----+--------------+------------+
|  1 | ugnamsung    | ES         |
|  2 | freakkid     | ES         |
|  3 | huangnanxuan | Chinese    |
+----+--------------+------------+
3 rows in set (0.00 sec)
```

INSERT 失败：

```
Database changed
mysql> INSERT INTO students (id, name, department) VALUES (NULL,"kid","history");

ERROR 1142 (42000): INSERT command denied to user 'huangnx'@'localhost' for table
 'students'
```

UPDATE 失败：

```
mysql> UPDATE studentS SET name="ugnamsung";
ERROR 1142 (42000): UPDATE command denied to user 'huangnx'@'localhost' for table
 'studentS'
```

DELETE 失败：

```
mysql> DELETE FROM students WHERE name='ugnamsung';
ERROR 1142 (42000): DELETE command denied to user 'huangnx'@'localhost' for table
 'students'
```

增加和删除列 失败：

```
mysql> ALTER TABLE students ADD age INT;
ERROR 1142 (42000): ALTER command denied to user 'huangnx'@'localhost' for table
'students'
mysql> ALTER TABLE students DROP name;
ERROR 1142 (42000): ALTER command denied to user 'huangnx'@'localhost' for table
'students'
```

删除 students 失败：

```
mysql> DROP TABLE students;
ERROR 1142 (42000): DROP command denied to user 'huangnx'@'localhost' for table '
students'
```

删除 test_tbl 失败：

```
mysql> DROP DATABASE test_tbl;
ERROR 1044 (42000): Access denied for user 'huangnx'@'localhost' to database 'tes
t_tbl'
```

综上可知，用户 huangnx 对 test_tbl 只有进行 select 操作的权限。

*** 登录用户 freakkid**

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| test_tbl           |
+--------------------+
5 rows in set (0.00 sec)

mysql> USE test_tbl;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW tables;
+--------------------+
| Tables_in_test_tbl |
+--------------------+
| courses            |
| students           |
+--------------------+
2 rows in set (0.00 sec)
```

用户 freakkid 可以看到 root 用户所有的 DATABASES，可看到 test_tbl 中的 students 和 courses

测试 freakkid 的权限：
SELECT 成功：

```
mysql> SELECT * FROM students;
+----+--------------+------------+
| id | name         | department |
+----+--------------+------------+
|  6 | ugnamsung    | ES         |
|  7 | freakkid     | ES         |
|  8 | huangnanxuan | Chinese    |
+----+--------------+------------+
3 rows in set (0.00 sec)
```

INSERT 成功：

```
mysql> INSERT INTO students (id, name, department) VALUES (NULL,"kid","history");

Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO students (id, name, department) VALUES (NULL,"kudo","CS");
Query OK, 1 row affected (0.06 sec)

mysql> SELECT * FROM students;
+----+-------------+------------+
| id | name        | department |
+----+-------------+------------+
|  6 | ugnamsung   | ES         |
|  7 | freakkid    | ES         |
|  8 | huangnanxuan | Chinese   |
|  9 | kid         | history    |
| 10 | kudo        | CS         |
+----+-------------+------------+
5 rows in set (0.00 sec)
```

UPDATE 成功：

```
mysql> UPDATE students SET department="ES" WHERE name="huangnanxuan";
Query OK, 1 row affected (0.06 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM students;
+----+-------------+------------+
| id | name        | department |
+----+-------------+------------+
|  6 | ugnamsung   | ES         |
|  7 | freakkid    | ES         |
|  8 | huangnanxuan | ES        |
|  9 | kid         | history    |
| 10 | kudo        | CS         |
+----+-------------+------------+
5 rows in set (0.00 sec)
```

DELETE 成功：

```
mysql> DELETE FROM students WHERE name='ugnamsung';
Query OK, 1 row affected (0.07 sec)

mysql> SELECT * FROM students;
+----+-------------+------------+
| id | name        | department |
+----+-------------+------------+
|  7 | freakkid    | ES         |
|  8 | huangnanxuan | ES        |
|  9 | kid         | history    |
| 10 | kudo        | CS         |
+----+-------------+------------+
4 rows in set (0.00 sec)
```

增加和删除列 成功：

```
mysql> ALTER TABLE students ADD age INT;
Query OK, 0 rows affected (0.67 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM students;
+----+--------------+------------+------+
| id | name         | department | age  |
+----+--------------+------------+------+
|  7 | freakkid     | ES         | NULL |
|  8 | huangnanxuan | ES         | NULL |
|  9 | kid          | history    | NULL |
| 10 | kudo         | CS         | NULL |
+----+--------------+------------+------+
```

```
mysql> ALTER TABLE students DROP age;
Query OK, 0 rows affected (0.66 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM students;
+----+--------------+------------+
| id | name         | department |
+----+--------------+------------+
|  7 | freakkid     | ES         |
|  8 | huangnanxuan | ES         |
|  9 | kid          | history    |
| 10 | kudo         | CS         |
+----+--------------+------------+
4 rows in set (0.00 sec)
```

创建 table test 成功：

```
mysql> CREATE TABLE test (id INT);
Query OK, 0 rows affected (0.26 sec)

mysql> SHOW tables;
+--------------------+
| Tables_in_test_tbl |
+--------------------+
| courses            |
| students           |
| test               |
+--------------------+
3 rows in set (0.00 sec)
```

删除 course 成功：

```
mysql> DROP TABLE courses;
Query OK, 0 rows affected (0.15 sec)

mysql> SHOW tables;
+--------------------+
| Tables_in_test_tbl |
+--------------------+
| students           |
| test               |
+--------------------+
2 rows in set (0.00 sec)
```

删除 test_tbl 成功：

```
mysql> DROP DATABASE test_tbl;
Query OK, 2 rows affected (0.38 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)
```

**3. 登录 root 用户，收回 root 分配给其他用户的权限：**

*** 回收 huangnx 对 test_tbl.students 的 SELECT 特权**

由于刚才在实验的时候删除了 test_tbl 数据库，现在重新创建并重新分配特权：

```
mysql> USE test_tbl;
Database changed
mysql> CREATE TABLE students (id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    -> name VARCHAR(20),
    -> department VARCHAR(30));
Query OK, 0 rows affected (0.25 sec)

mysql> GRANT SELECT
    -> ON test_tbl.students
    -> TO 'huangnx'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> show grants for "huangnx"@"localhost";
+-----------------------------------------------------------------+
| Grants for huangnx@localhost                                    |
+-----------------------------------------------------------------+
| GRANT USAGE ON *.* TO 'huangnx'@'localhost'                     |
| GRANT SELECT ON `test_tbl`.`students` TO 'huangnx'@'localhost' |
+-----------------------------------------------------------------+
2 rows in set (0.00 sec)
```

回收 SELECT 特权：

```
mysql>  revoke select on test_tbl.students from  'huangnx'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> show grants for "huangnx"@"localhost";
+-----------------------------------------------+
| Grants for huangnx@localhost                  |
+-----------------------------------------------+
| GRANT USAGE ON *.* TO 'huangnx'@'localhost' |
+-----------------------------------------------+
1 row in set (0.00 sec)
```

登录 huangnx:

```
nanxuan@nanxuan-Lenovo-G40-70m:~$ mysql -u huangnx -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 5.7.17-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2016, 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> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
+--------------------+
1 row in set (0.00 sec)
```

可以看到权限已经被 root 收回，huangnx 不能查看任何 root 创建的 DATABASE

*** 回收 huangnx 对 test_tbl.students 的 SELECT 特权**

```
mysql> revoke all privileges, grant option from "freakkid"@"localhost";
Query OK, 0 rows affected (0.00 sec)

mysql> show grants for "freakkid"@"localhost";
+-----------------------------------------------+
| Grants for freakkid@localhost                 |
+-----------------------------------------------+
| GRANT USAGE ON *.* TO 'freakkid'@'localhost' |
+-----------------------------------------------+
1 row in set (0.00 sec)
```

登录 freakkid：

```
nanxuan@nanxuan-Lenovo-G40-70m:~$ mysql -u freakkid -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.7.17-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2016, 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> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
+--------------------+
1 row in set (0.00 sec)

mysql>
```

可以看到权限已经被 root 收回，freakkid 不能查看任何 root 创建的 DATABASE。

## 二、实验遇到的问题和解决方案

1.创建用户时用了 INSERT 语句创建的，出现了报错

```
mysql> INSERT INTO user
    ->  (user, host, authentication_string)
    -> VALUES ('sharing', 'localhost', PASSWORD('guset*ABC*1000'));
ERROR 1364 (HY000): Field 'ssl_cipher' doesn't have a default value
```

问了谷歌，原来是为了安全性所以 MYSQL 默认不允许用 INSERT 创建用户，解决方法就是更改配置文件 my.cnf 即可，但我觉得 MYSQL 做的很有道理，于是就放弃更改配置文件，使用 create 语句创建用户。

2.进行实验前，先做了一些准备，在网上找到一个 MYSQL 的教程边看边实验，设计实验流程，再开始进行实验，由于在实验前学习 sql 语句时创建过用户 ugnamsung 并删除，但在正式实验中再次创建 ugnamsung 同名用户时却出现报错：

```
mysql> create user 'ugnamsung'@'localhost' identified by 'guset*ABC*1000';
ERROR 1396 (HY000): Operation CREATE USER failed for 'ugnamsung'@'localhost'
```

查询了已经创建的 user 并没有发现有同名的用户的存在：

```
mysql> select user from mysql.user;
+------------------+
| user             |
+------------------+
| debian-sys-maint |
| mysql.sys        |
| root             |
+------------------+
3 rows in set (0.00 sec)
```

最后在网上搜到了一个解决方法，输入
FLUSH PRIVILEGES;
再创建 ugnamsung 就可以了。

网上有说的是 MYSQL 的一个 bug，与特权有关的更改未能及时更新而造成的。不过这样很容易造成滥用 FLUSH PRIVILEGES;后面有什么错误无法创建用户或者无法给用户分配特权，都会试试看用 FLUSH PRIVILEGES;会不会有奇效而不是检查自己的语句有没有写错。

2. 在给 freakkkid 分配特权的时候出现错误

```
mysql> USE mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> grant all privileges on *.* to 'freakkid@localhost';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirement
s
```

看了一下报错，发现是当初安装 mysql 的时候一不小心把密码安全性认证设置得太高了，可是又觉得 freakkid 用户的密码复杂度应该符合它的要求了，觉得很奇怪，但还是把密码复杂度调低了：

```
mysql> SHOW VARIABLES LIKE 'validate_password%';
+--------------------------------------+--------+
| Variable_name                        | Value  |
+--------------------------------------+--------+
| validate_password_check_user_name    | OFF    |
| validate_password_dictionary_file    |        |
| validate_password_length             | 8      |
| validate_password_mixed_case_count   | 1      |
| validate_password_number_count       | 1      |
| validate_password_policy             | MEDIUM |
| validate_password_special_char_count | 1      |
+--------------------------------------+--------+
7 rows in set (0.01 sec)
```

```
mysql>  set global validate_password_policy=0;
Query OK, 0 rows affected (0.00 sec)
```

调低密码安全性认证后，还是无法给 freakkid 分配特权，在网上搜了一些方法（包括使用了 FLUSH PRIVILEGES;），屡试无效后，发现是我分配特权的语句写错了。实在是太粗心了。

## 三、实验总结和反思

挺喜欢 SQL 这个关系数据库管理工具，觉得很好玩，sql 大多数语句它都能实现，自带很多实用的函数（包括数学和日期方面），很强大，并且是开源的，可以免费使用。
由于不喜欢单纯的输入语句并验证，所以就选择了这个可以创建用户分配权限的实验，遇到一些问题并解决。

编程要求程序员在写代码的时候尽量减少语法错误，而不是在编译报错后再一个个回头解决问题，浪费时间，降低效率。我想 sql 也一样，输入语句的时候还是保证语法正确性，注意标点符号等细节，就不用花太多时间在解决低级语法错误并重新输入上，但在解决问题的过程让我知道了 mysql 一些配置文件、密码安全性认证修改、设置用户权限等，正所谓在踩坑中成长。