浙江大学

数据库系统实验报告

作业名称：	SQL 安全性
姓　　名：	汪珉凯
学　　号：	3220100975
电子邮箱：	3220100975@zju.edu.cn
联系电话：	18157421318
指导老师：	孙建伶

2024 年 3 月 26 日

实验名称

一、实验目的

熟悉通过 SQL 进行数据完整性控制的方法。

二、实验环境

MySQL

三、实验流程

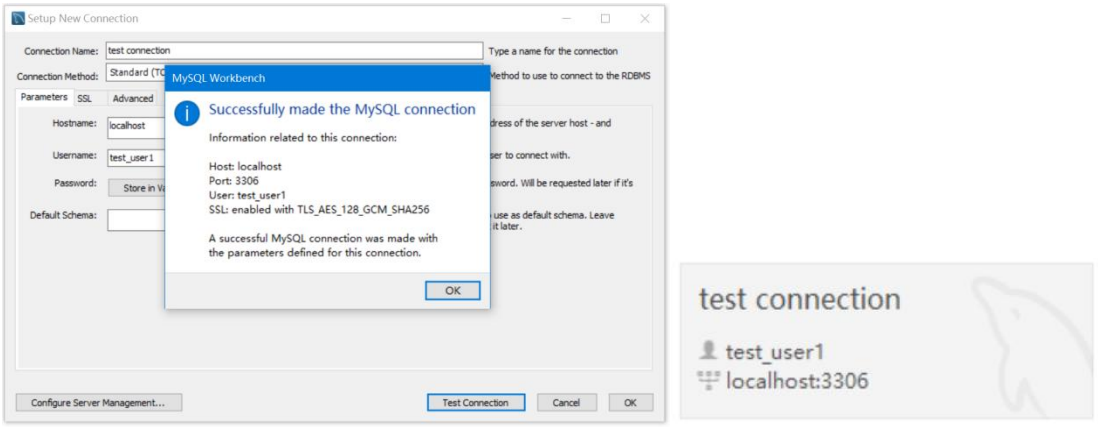
0.实验准备环节

0.1 以 root 身份登录 MySQL，输入如下代码，创建一个新的用户：

```
1 •  CREATE USER 'test_user1'@'localhost' IDENTIFIED BY 'wmk@000056';
```

得到如下结果，说明新用户创建成功：

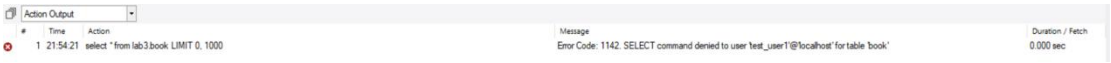
0.2 以 test_user1 的身份登陆数据库：



0.3 尝试以 test_user1 的身份访问 lab3 的 book 表
执行以下代码：

```
1 •  select * from lab3.book;
```

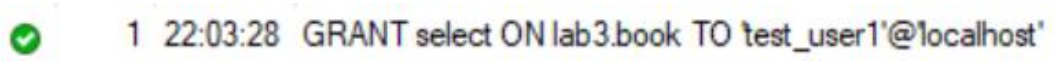
但结果发现新用户无法访问 lab3 数据库下的 book 表：



0.4 回到 root 身份，给新用户授权：

```
1 •  GRANT select ON lab3.book TO 'test_user1'@'localhost';
```

结果如下：



0.5 查看新用户的权限
执行以下代码：

```
1 •  show grants for 'test_user1'@'localhost';
```

得到结果如图：

Grants for test_user1@localhost
GRANT USAGE ON *.* TO `test_user1`@`localhost`
GRANT SELECT ON `lab3`.`book` TO `test_user1`@`localhost`

0.6 查看有哪些用户在特定表上有哪些权限
执行以下代码：

```
1 •  select * from mysql.tables_priv where table_name='book';
```

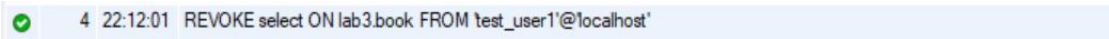
得到如图结果：

Host	Db	User	Table_name	Grantor	Timestamp	Table_priv	Column_priv
localhost	lab3	test_user1	book	root@localhost	2024-03-25 22:03:28	Select	
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

0.7 收入赋予新用户的权限
执行以下代码：

```
1 •  REVOKE select ON lab3.book FROM 'test_user1'@'localhost';
```

得到以下结果：



1.考察表的生成者拥有该表的哪些权限。

1.1 首先以 root 身份执行以下代码，将在 lab3 数据库新建表的权限赋予临时用户 test_user1：

```
1 •  GRANT create ON lab3.* TO 'test_user1'@'localhost';
```

1.2 以 test_user1 的身份执行以下代码，在 lab3 数据库上建立新表：

```
use lab3;
create table user1_table1(
    var1 int,
    var2 varchar(20),
    primary key (var1)
);
```

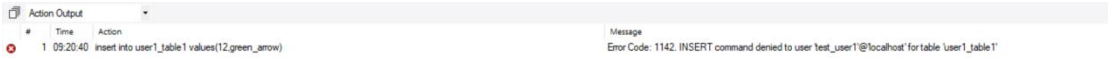
回到 root 身份，可以看到 lab3 数据库中多了一张表：

1.3 尝试以新用户身份向新建表中插入数据

执行以下代码：

```
insert into user1_table1 values(12,green_arrow);
```

得到如下结果：



09:20:40 insert into user1_table1 values(12,green_arrow) Error Code: 1142. INSERT command denied to user 'test_user1'@'localhost' for table 'user1_table1' 0.000 sec

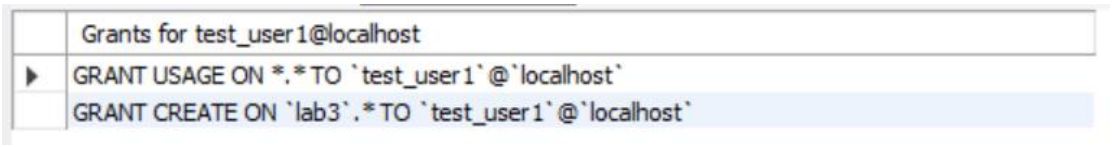
由报错信息可以看到，插入数据指令被拒绝，原因是新用户仅有创建新表的权力，但没有修改新表的权力。

1.4 以 root 身份，查看新用户的所有权限：

执行以下代码：

```
show grants for 'test_user1'@'localhost';
```

得到结果如图：



可以看到新用户在 lab3 数据库上仅有创建新表的权限。

2. 考察 grant 和 revoke 命令对其他用户进行授权和权力回收的作用。

2.1 将向 user1_table1 表中插入数据的权限赋予新用户

执行以下代码：

```
1 •  grant select on lab3.user1_table1 to 'test_user1'@'localhost' ;
2 •  grant insert on lab3.user1_table1 to 'test_user1'@'localhost' ;
```

得到如下结果：

以 test_user1 的身份运行以下代码，尝试向 user1_table1 表格中插入数据：

```
1 •  use lab3;
2 •  insert into user1_table1 values
3    (1,'flash'),
4    (2,'green_arrow'),
5    (3,'batman'),
6    (4,'joker'),
7    (5,'shark_king')
8    ;
9 •  select * from user1_table1;
```

得到结果如下：

	var1	var2
▶	1	flash
	2	green_arrow
	3	batman
	4	joker
	5	shark_king
*	NULL	NULL

由此可见，在赋予新用户插入数据的权限后，他就可以向规定表中插入数据。

2.2 将更新 user1_table1 表中数据的权限赋予新用户

以 root 身份执行以下代码：

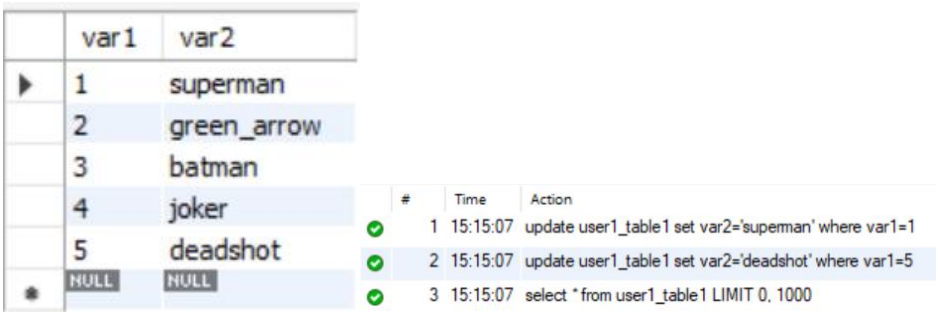
```
1    grant update on lab3.user1_table1 to 'test_user1'@'localhost';
```

得到如下结果：

以 test_user1 的身份运行以下代码，尝试更新 user1_table1 表格中的数据：

```
update user1_table1 set var2='superman' where var1=1;
update user1_table1 set var2='deadshot' where var1=5;
select * from user1_table1;
```

得到结果如下，表中数据的确得到了更新：



	var1	var2
▶	1	superman
	2	green_arrow
	3	batman
	4	joker
	5	deadshot
*	NULL	NULL

#	Time	Action
1	15:15:07	update user1_table1 set var2='superman' where var1=1
2	15:15:07	update user1_table1 set var2='deadshot' where var1=5
3	15:15:07	select * from user1_table1 LIMIT 0, 1000

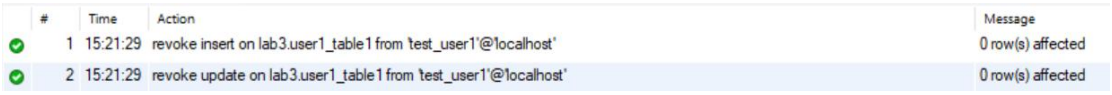
由此可见，在赋予新用户更新数据的权限后，他就可以更新规定表中的数据。

2.3 收回赋予新用户的插入权限和更新权限

以 root 身份执行以下代码：

```
revoke insert on lab3.user1_table1 from 'test_user1'@'localhost';
revoke update on lab3.user1_table1 from 'test_user1'@'localhost';
```

得到如下结果：



#	Time	Action	Message
1	15:21:29	revoke insert on lab3.user1_table1 from 'test_user1'@'localhost'	0 row(s) affected
2	15:21:29	revoke update on lab3.user1_table1 from 'test_user1'@'localhost'	0 row(s) affected

以 test_user1 的身份运行以下代码，尝试插入向 user1_table1 表中插入数据：

```
insert into user1_table1 values(6,'aquman');
```

得到结果如下：

#	Time	Action	Message
1	15:24:24	insert into user1_table1 values(6,'aquman')	Error Code: 1142. INSERT command denied to user 'test_user1'@'localhost' for table 'user1_table1'

15:24:24 insert into user1_table1 values(6,'aquman') Error Code: 1142. INSERT command denied to user 'test_user1'@'localhost' for table 'user1_table1' 0.000 sec

以 test_user1 的身份运行以下代码，尝试更新 user1_table1 表格中的数据：

```
update user1_table1 set var2='superman' where var1=1;
update user1_table1 set var2='deadshot' where var1=5;
```

得到结果如下：

15:25:48 update user1_table1 set var2='superman' where var1=1 Error Code: 1142. UPDATE command denied to user 'test_user1'@'localhost' for table 'user1_table1' 0.000 sec

由此可见：此时 revoke 指令已经成功将原先赋予新用户的权限进行了收回。

3.考察通过视图进行权限控制的作用。

3.0 为了使实验目的更好地得到体现，首先以 root 身份对原表格进行补充。

```
alter table user1_table1 add var3 varchar(20);
update user1_table1 set var3='inevitable' where var1 = 1;
update user1_table1 set var3='rich' where var1 = 2;
update user1_table1 set var3='rich' where var1 = 3;
update user1_table1 set var3='wisdom' where var1 = 4;
update user1_table1 set var3='shot' where var1 = 5;
```

得到结果如图，成功更新：

	var1	var2	var3
▶	1	superman	inevitable
	2	green_arrow	rich
	3	batman	rich
	4	joker	wisdom
	5	deadshot	shot
*	NULL	NULL	NULL

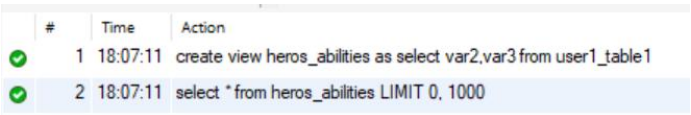
3.1 以 root 身份创建 view

执行以下代码：

```
create view heros_abilities as
select var2,var3 from user1_table1;
select * from heros_abilities;
```

结果如下所示：

var2	var3
superman	inevitable
green_arrow	rich
batman	rich
joker	wisdom
deadshot	shot



3.2 将该视图的查询权限赋予新用户，考察其作用。
执行以下代码：

```
1 • grant select on heros_abilities to 'test_user1'@'localhost';
```

得到如下结果：

然后以新用户的身份查看该视图：

```
1 • select * from heros_abilities;
```

得到结果如图：

	var2	var3
▶	superman	inevitable
	green_arrow	rich
	batman	rich
	joker	wisdom
	deadshot	shot

可以看到新用户能够成功查询该视图。

3.3 将该视图的更新权限赋予新用户，考察其作用。
执行以下代码：

```
1 • grant update on heros_abilities to 'test_user1'@'localhost';
```

得到如下结果：

然后以新用户的身份更新该视图并查看：

```
1 • update heros_abilities set var3='super rich' where var2 = 'green_arrow';
2 • select * from heros_abilities;
```

得到结果如图：

	var2	var3
▶	superman	inevitable
	green_arrow	super rich
	batman	rich
	joker	wisdom
	deadshot	shot

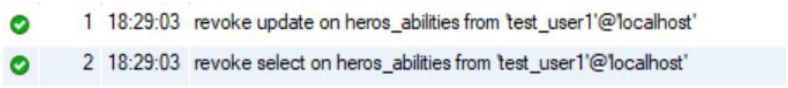
可以看到新用户能够成功更新该视图。

3.4 收回新用户对该视图的权限。

以 root 身份执行以下代码：

```
revoke update on heros_abilities from 'test_user1'@'localhost';
revoke select on heros_abilities from 'test_user1'@'localhost';
```

结果如图所示：



以新用户的身份尝试查询该视图，得到如下结果：

18:30:17 select * from heros_abilities LIMIT 0, 1000 Error Code: 1142. SELECT command denied to user 'test_user1'@'localhost' for table 'heros_abilities' 0.000 sec

以新用户的身份尝试更新该视图，得到如下结果：

18:31:08 update heros_abilities set var3='super rich' where var2 = 'green_arrow' Error Code: 1142. UPDATE command denied to user 'test_user1'@'localhost' for table 'heros_abilities' 0.000 sec

四、遇到的问题及解决方法

本次实验比较简单，基本没有遇到什么问题。只是有时候会把 sql 指令弄混，只需要参考一下 MySQL 的操作手册就可以解决。

五、总结

这次实验相较于前三次比较综合，希望多加练习，以后能清楚无误地写出各类操作，不再需要借助任何外力。