

浙江大学

数据库系统实验报告

作业名称:	SQL 安全性
姓 名:	龙永奇
学 号:	3220105907
电子邮箱:	3220105907@zju.edu.cn
联系电话:	15393113093
指导老师:	孙建伶

2024 年 03 月 28 日

SQL 安全性

一、实验目的

1. 熟悉通过 SQL 进行数据安全性控制的方法

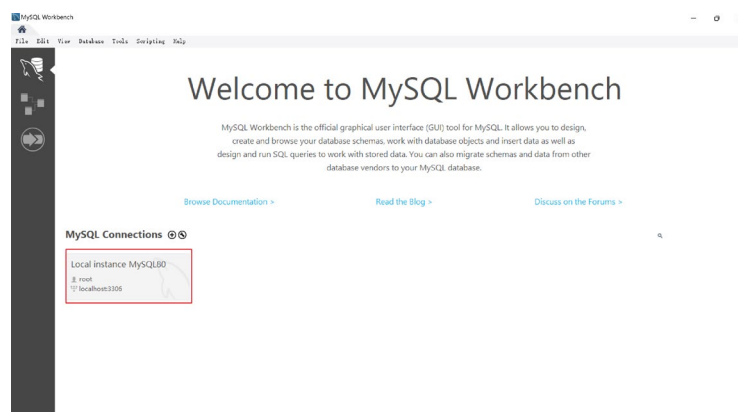
二、实验环境

1. 数据库管理系统：MySQL
2. 操作系统：Windows11 企业版

三、实验流程

1. 权限管理

使用 root 身份登录 MySQL:



root 用户拥有全部权限，由 root 用户为普通用户分配权限
新建普通用户 proton:

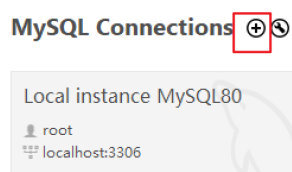
```
CREATE USER "proton"@"%" IDENTIFIED BY'153351';
```

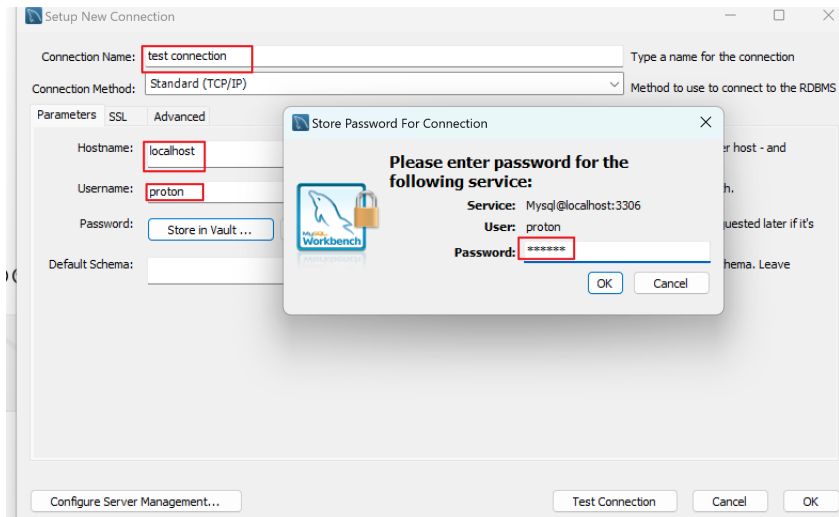
其中 ‘%’ 表示对所用主机开放权限，在这里我们使用 localhost:

```
CREATE USER "proton"@"localhost" IDENTIFIED BY'153351';
```

✓	9	22:05:10	SELECT user FROM mysql.user LIMIT 0, 1000	4 row(s) returned
✓	10	22:05:43	CREATE USER 'proton02'@'localhost' IDENTIFIED BY 'bamboo'	0 row(s) affected

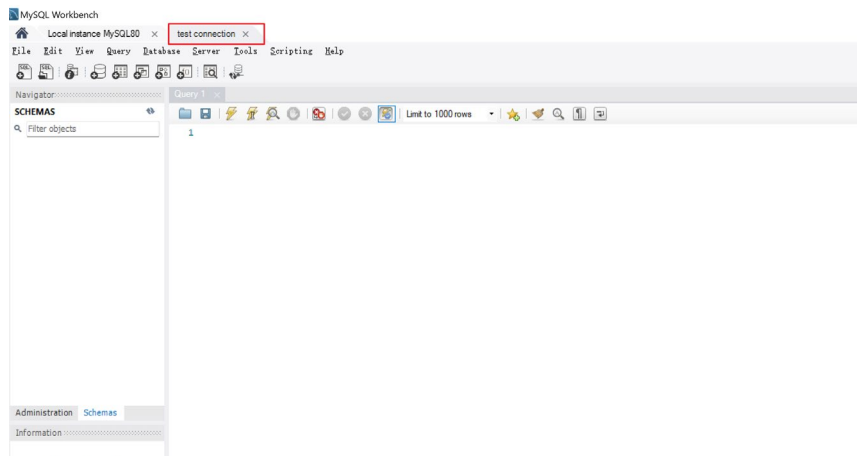
使用用户 proton 登录 MySQL:





（此时遇到无法登录的问题，详见第四部分）

登陆成功：



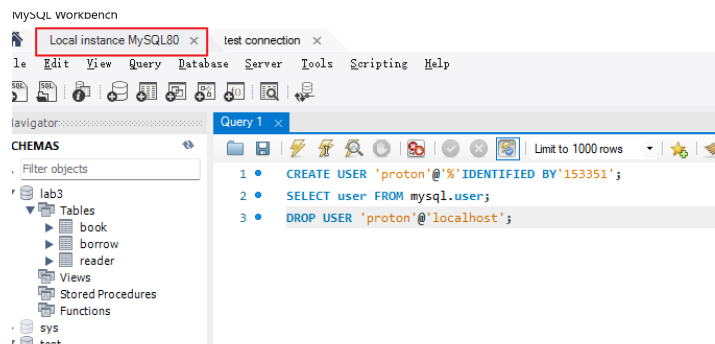
尝试访问 root 用户建立的表：

```
SELECT * FROM test.company;
```

出现错误：

#	Time	Action	Message
1	22:46:38	select * from test.company LIMIT 0, 1000	Error Code: 1142. SELECT command denied to user 'proton'@'localhost' for table 'company'

说明此用户无法访问 root 用户建立的表。返回 root 用户：



为新建用户授权：

```
GRANT SELECT ON test.company TO 'proton'@'%';
```

授权成功:

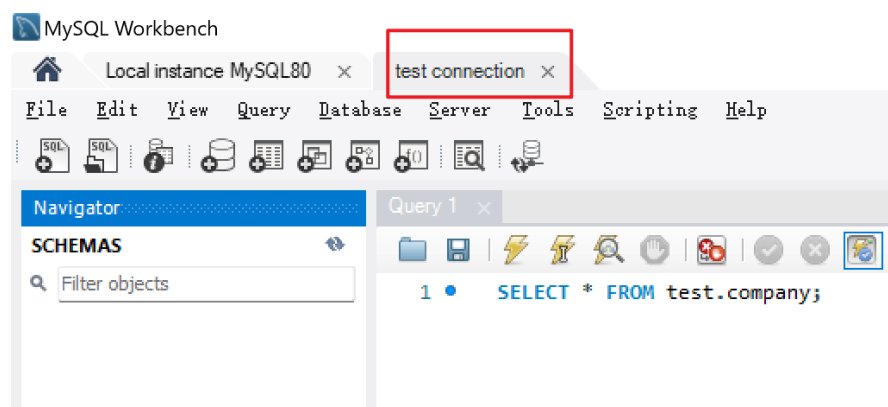
✓	2	22:36:16	DROP USER 'proton02'@'localhost'
✓	3	22:36:35	CREATE USER 'proton'@'% ' IDENTIFIED BY '153351'
✓	4	22:36:42	SELECT user FROM mysql.user LIMIT 0, 1000
✓	5	23:29:51	GRANT SELECT ON test.company TO 'proton'@'% ';

查看不同用户对于 company 的权限:

```
SELECT * FROM mysql.tables_priv  
WHERE table_name = 'company';
```

Result Grid									
		Filter Rows:		Edit:		Export/Import:		Wrap Cell Con	
	Host	Db	User	Table_name	Grantor	Timestamp	Table_priv	Column_priv	
	%	test	proton	company	root@localhost	2024-03-27 23:29:51	Select		
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

返回用户 proton:



再次查询 company:

Result Grid			
		Filter Rows:	
	company_id	company_name	city
▶	1	Small Bank Corporation	New York
	2	Small Bank Corporation	Los Angeles
	3	Small Bank Corporation	Chicago
	4	deutsche Bank Corporation	New York
	5	deutsche Bank Corporation	Los Angeles
	6	deutsche Bank Corporation	Chicago
	7	Global Corp	Chicago
*	NULL	NULL	NULL

此时用户 proton 可以查看用户 root 创建的表

尝试收回权限:

```
REVOKE SELECT ON test.company  
FROM 'proton'@'%';
```

回收成功:

✓	11	00:01:07	SELECT * FROM company LIMIT 0, 1000	7 row(s) returned
✓	12	00:02:29	REVOKE SELECT ON test.company FROM 'proton'@'% ';	0 row(s) affected

再次尝试查询：

7 00:09:26 SELECT * FROM company LIMIT 0, 10... 7 row(s) returned

8 00:09:40 SELECT * FROM company LIMIT 0, 10... Error Code: 1142. SELECT command denied to user 'proton'@'localhost' for table 'company'

访问被拒绝

2. 建立表，考察表的生成者拥有该表的哪些权限

尝试查看普通用户 proton 所建立的表拥有哪些权限

通过 root 用户为 proton 用户授予在数据库 test 上建立表格的权限：

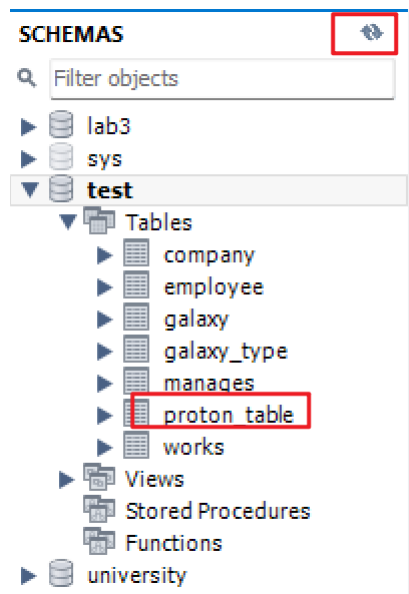
```
GRANT CREATE ON test.* TO 'proton'@'%';
```

16 00:23:55 GRANT CREATE ON test.* TO 'proton'@'%' 0 row(s) affected

使用用户 proton 建立表格 proton_table：

```
CREATE TABLE proton_table(  
    id CHAR(20),  
  
    PRIMARY KEY(id)  
);
```

在 root 用户界面刷新：



成功建立表格

对 proton 在 proton_table 授予其他权限：

```
GRANT SELECT ON test.proton_table TO 'proton'@'%';  
GRANT UPDATE ON test.proton_table TO 'proton'@'%';  
GRANT INSERT ON test.proton_table TO 'proton'@'%';  
GRANT DELETE ON test.proton_table TO 'proton'@'%';
```

12 00:43:56 CREATE TABLE proton_table(id CHAR(20), PRIMA... 0 row(s) affected

3. 使用 SQL 的 `grant` 和 `revoke` 命令对其他用户进行授权和权限回收，考察相应的作用

详见步骤 1

4. 建立视图，并把该视图的查询权限授予其他用户，考察通过视图进行权限控制的作用

通过 root 用户建立视图 `proton_window`:

```
CREATE VIEW proton_window AS
SELECT NGC, distance
FROM galaxy
```

22 00:57:57 CREATE VIEW proton_window AS SELECT NGC, distance FROM galaxy 0 row(s) affected

授予用户 `proton` 该视图的查询权限:

```
GRANT SELECT ON test.proton_window TO 'proton'@'%';
```

23 01:01:26 GRANT SELECT ON test.proton_window TO 'proton'@'%' 0 row(s) affected

查询:

```
SELECT *
FROM proton_window;
```

Result Grid			Filter
	NGC	distance	
▶	224	7100	
	1952	7200	
	5272	32200	
	7089	36900	

可以看到 `proton` 拥有查询该视图的权限

四、遇到的问题及解决方法

1. 在第一步建立 test connection 过程中，输入预设的密码后会出现：



在查询后通过以下方法解决：

该问题原因是此时启动的是 windows 下注册的服务，但系统并没有注册 MySQL 到服务中，即当前路径下没有 MySQL 服务。

首先进入 MySQL 中的 bin：

```
PS D:\Code\MySQL> cd bin
PS D:\Code\MySQL\bin>
```

输入 `mysqld -install`：

```
PS D:\Code\MySQL\bin> mysqld --install
Service successfully installed.
```

执行 `net start mysql`：

```
PS D:\Code\MySQL\bin> net start mysql
MySQL 服务正在启动 .
MySQL 服务无法启动。

服务没有报告任何错误。

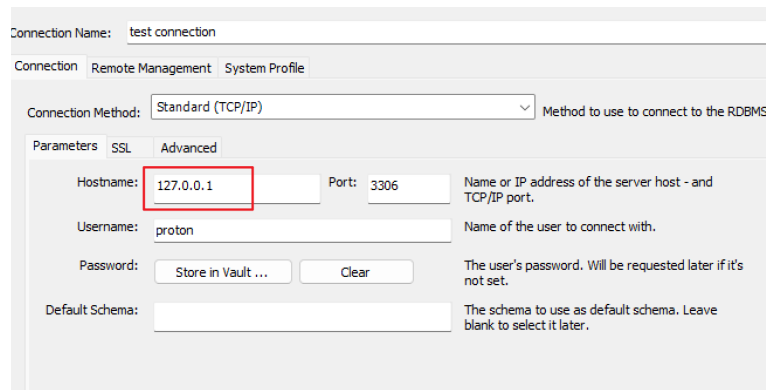
请键入 NET HELPMSG 3534 以获得更多的帮助。
```

重新执行 `mysql -initialize`：

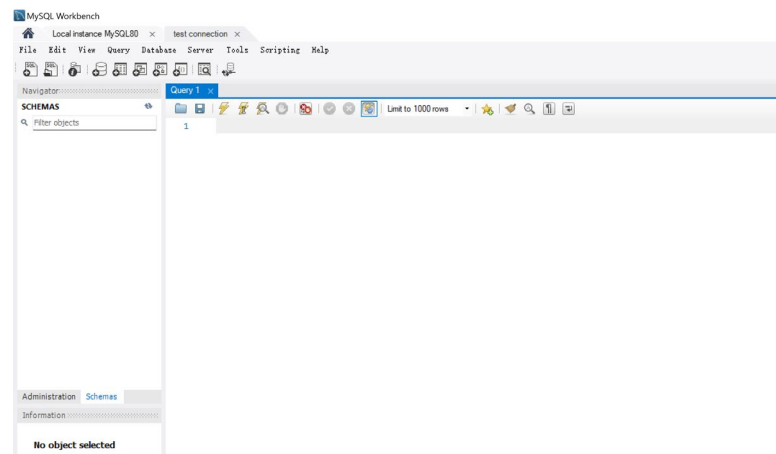
```
PS D:\Code\MySQL\bin> mysqld --initialize
PS D:\Code\MySQL\bin>
```

但仍然出现相同问题，这里不求甚解

随后在创建用户时将 hostname 置为 %，尝试修改 hostname：



最终创建成功：



五、总结

在本次实验中，通过使用 MySQL 数据库管理系统，进行了数据安全性控制的实验。通过 root 用户登录 MySQL，使用 root 用户为普通用户分配了权限。创建普通用户时，遇到了无法登录的问题，但通过修改主机名为`%`解决。同时使用了 SQL 的 GRANT 和 REVOKE 命令为普通用户分配对特定表以及视图的权限，并能够控制权限的收回与授予。

通过本次实验，进一步了解了如何使用 SQL 进行数据安全性控制，将理论知识用于实践，同时注意到理论课程中使用到的语法和 MySQL 等具体数据库的语法有所不同，总体而言收获颇丰，下次实验继续努力！