# 实验六：数据库的安全性

## 一、实验目的

- 理解数据库系统用户(user)、权限(privilege)和角色(role)的概念和作用
- 熟练掌握用户的管理：用户的创建、查看和删除
- 熟练掌握角色的管理：角色的创建、查看和删除
- 熟练掌握使用 GRANT 命令给用户或角色授权的方法，包括权限的转授（WITH GRANT OPTION）
- 熟练掌握使用 REVOKE 命令将权限回收的方法
- 熟练掌握修改用户权限或角色权限的方法
- 熟练掌握查看用户列表、权限列表的方法
- 理解视图的安全性作用

## 二、实验要求

- 设计正确的 SQL 语句并测试其是否实现了安全性要求
- 完成实验内容，撰写并提交实验报告（按实验报告模板）
- 实验报告提交截止日期：2025 年 5 月 11 日 00:00（星期日）
- **选项：**自行完成教材上的例题，但无需放入实验报告中

## 三、实验内容与步骤

因本实验涉及多用户的切换，为简化说明，在命令前使用符号 user_name>表示当前用户为 user_name，该符号并不是命令的组成部分

**1.创建新用户**

(1)以 root 身份登录到系统，发布查询语句：**root>SELECT user FROM mysql.user;** 记录查询结果

(2) 创建两个新用户：**Alice** 和 **Bob**

**root>CREATE USER Alice@localhost IDENTIFIED BY 'root';** *--口令简化，可自定*

**root>CREATE USER Bob@localhost IDENTIFIED BY 'root';** *--同上*

**2.查看用户列表及其权限**

**方式一：通过查询 mysql 数据库的 user 表**

Mysql8.4 将把所有用户的信息放在系统自带的 mysql 数据库的 user 表中，所以查询该表能够得到用户及其权限信息。

(1)查看 user 表的结构

命令：**root>DESC mysql.user;**

user 表有 51 个字段，所有以 priv 结尾的字段名均为用户是否具有该字段对应的权限（值为 'Y' 或 'N'）

(2)查看用户列表

命令：**root>SELECT user FROM mysql.user;** *--检查 Alice 和 Bob 是否出现在查询结果中*

- 使用命令 SELECT user(); 可以查询当前用户名，这对不记得当前用户是谁非常有用

(3)查询所有用户的权限

命令：**root>SELECT * FROM mysql.user;**

(4)查询特定用户 Alice 的权限

命令：**root>SELECT * FROM mysql.user WHERE user='Alice';**

注：查询其他用户权限只需将 Alice 改为待查询用户名，如 Bob

**方式二：使用 SHOW 命令查询用户权限**

可以使用 SHOW 命令显示当前用户或特定用户的权限

(5)查询当前用户的权限

任何一个用户均可以使用以下命令中的任何一条来查询自己的权限：

**SHOW grants;**
**SHOW grants for current_user;**
**Show grants for current_user();**

(6)查询特定用户的权限

命令：**root>show grants for user_name@localhost;**

如，查询 Alice 的权限：**root>show grants for Alice@localhost;**

**注意**上述命令中**用户名格式的正确使用**。完整的用户名由三部分组成：user_name, @, localhost。当

任何用到用户名时一般要保证用户名格式必须与创建时的格式等同，否则系统易报错，如：

创建新用户格式：**root>create user caesar@localhost identified by 'root';**

查询用户权限的用户格式必须为：**root>show grants for caesar@localhost;**

如果使用命令：**root>show grants for caesar;** 则报错，因为此时的 caesar 默认为 caesar@'%'，

而不是 caesar@localhost。同理，如果新用户创建命令为：**root>create user caesar identified by 'root'**；查询权限时如果使用：**root>show grants for caesar@localhost**；系统也将报错。

**3.查看系统的所有权限**

任何一个用户都可以发布命令：**show privileges;** 来显示系统的所有权限及其作用的对象级别，共有 73 条记录。Server Admin-服务器级别，Databases-数据库级别，Tables-表级别

**•USAGE 权限**

USAGE 权限是一种特殊的权限，任何一个新建用户都将被自动授予 USAGE 权限。拥有 USAGE 权限的用户将被允许连接到 mysql 数据库，但无操作数据库的能力。

**4.创建视图**

(1)以实验二中自己中文名的拼音用户身份（以下均以 user1 替代）登录 mysql 系统，进入 sales 数据库

(2)创建视图 salesman，该视图只保存 employees 表中所有 job_title 为'Sales Representative'的雇员

(3)查询视图 salesman

**注：为方便观察后面的实验效果，该窗口始终不关闭**

**5.用户授权与权限回收**

(1)新开一个 CMD 窗口，以 Alice 身份登录系统，发布查询语句：

**Alice>SELECT * FROM sales.salesman;**

观察结果

(2)用户 user1 将查看 sales.salesman 的权限授予给 Alice，命令：

**user1>grant select on sales.salesman to Alice@localhost;**

**user1>flush privileges;** *--该命令表明重新加载权限表，确保权限修改立即生效*

(3) 用户 user1 发布语句：**user1>show full tables;**

*--该命令的目的是为了显示选项 FuLL 的效果：结果出现表的类型，view 或 base tables*

(4)用户 Alice 重新发布语句：**Alice>SELECT * FROM sales.salesman;**

观察结果并与第(1)步结果比较，理解差异背后的原因

(5)Alice 发布语句：**Alice>SELECT * FROM sales.salesman where job_title='President';**

试解释该结果

(6)新开一个 CMD 窗口，以用户 Bob 登录系统，发布查询语句：

**Bob>SELECT * FROM sales.salesman;**

观察结果

(7)user1 用户回收 Alice 查看 sales.salesman 的权限，命令：

**user1>revoke select on sales.salesman from Alice@localhost;**
**user1>flush privileges;**

(8)用户 Alice 重新发布语句：**Alice>SELECT * FROM sales.salesman;** 观察结果

(9)用户 user1 将查看 sales.salesman 的权限授予给 Alice 并允许他将该权限转授给用户 Bob，命令：

**user1>grant select on sales.salesman to Alice@localhost with grant option;**

(10)用户 Alice 发布授权给 Bob 语句：

**Alice>grant select sales.salesman to Bob@localhost;**

(11)用户 Bob 发布查询语句：

**Bob>SELECT * FROM sales.salesman;**

观察系统反馈结果

**注意：**Mysql8 不支持级联回收权限，在本例中，在 user1 授权给 Alice 时使用了 with grant option 选项且 Alice 又把权限转授给 Bob 后，Bob 可以进行查询 Salesman 的操作。如果 user1 想使用 Cascade

来级联回收 Bob 从 Alice 那里获得的权限，那么 mysql8 是不支持的。要达成此任务的一种方式是：user1 直接从 Alice 那里回收权限（此时 Bob 获得的权限仍然有效），然后 Alice 直接从 Bob 那里回收权限（两个操作的次序不限）

(12)user1 回收授予给 Alice 的权限，Alice 回收授予给 Bob 的权限（这步为角色操作做准备）

**6.角色的创建与权限管理**

(1)用户 user1 创建一个名为 salesman_role 的角色

```
user1> create role salesman_role;
```

**注：**角色名的构成与用户名一样，不带@localhost 的角色名默认为 role_name@%，后续使用时的格式与创建时等同

(2)查看 salesman_role 角色包含的权限

```
user1> show grants for salesman_role; --将查询用户权限命令中的用户名改为角色名
```

(3)将 select 和 update 视图 Salesman 的权限授予给 salesman_role 角色

```
user1> grant selec, update on sales.salesman to salesman_role;
user1> flush privileges;
```

(4)查看 salesman_role 角色包含的权限，并与(2)比较

```
user1> show grants for salesman_role;
```

(5)回收 salesman_role 角色中的 update 权限

```
user1> revoke update on sales.salesman from salesman_role;
user1> flush privileges;
```

(6)查看 salesman_role 角色包含的权限，并与(4)比较

```
user1> show grants for salesman_role;
```

**说明：步骤(3)和(5)即为角色权限的修改操作**

(7)将角色授权给用户

```
user1> grant salesman_role to Alice@localhost;
user1> flush privileges;
```

**注**：将角色授权给用户，没有 ON dabase_name.table_name；且用户名格式等同创建时格式

(8)检验角色授权是否生效

```
Alice> select * from sales.salesman;
```

系统可能提示以下错误：

ERROR 1142 (42000): SELECT command denied to user 'Alice'@'localhost' for table 'salesman'

**原因**：mysql 中角色使用先设定是否启用，本实验仅介绍**会话级角色**的启用与关闭

```
Alice> set role salesman_role; 或者 Alice> set role all;

Alice> select * from sales.salesman;
```

关闭当前会话的生效角色

```
Alice> set role none;
```

(9)删除角色

```
user1> drop role salesman_role;
```

(10)删除用户

```
root> drop user Alice@localhost;
root> drop user Bob@localhost;
```

## 四、实验思考

在 Mysql8.4 中，

1.用户名和角色名的格式由那几部分组成？如何正确地使用用户名和角色名？

2. 步骤 5 之(5)的结果与视图有关吗？试解释之。

3.用户可以使用 with grant option 选项进行权限的转授，但在转授成功后能否使用 CASCADE 进行权限的级联回收，试举本实验例子说明。

**注：**将问题的解答放到实验报告的实验总结部分