

数据库系统课程实验报告

|  |  |
| --- | --- |
| 实验名称： | 数据库的安全性 |
| 实验日期： | 4.24 |
| 实验地点： | 西部片区4号楼103 |
| 提交日期： | 5.9 |
|  | |
| 学号： | 37220232203786 |
| 姓名： | 潘腾凯 |
| 专业年级： | 软工2023级 |
| 学年学期： | 2024-2025学年第二学期 |

1. 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

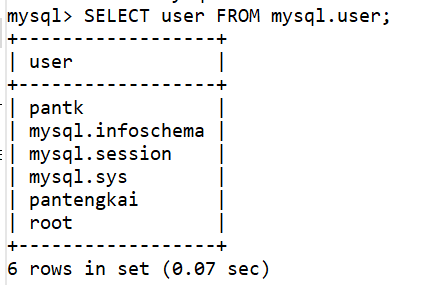
• 理解视图的安全性作用

1. 实验内容和步骤

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

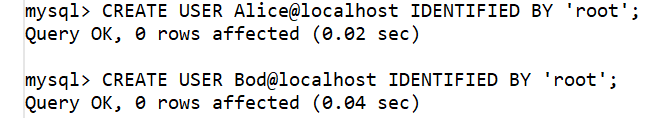
(1)以 root 身份登录到系统， 发布查询语句： root>SELECT user FROM mysql.user; 记录查询结果



(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

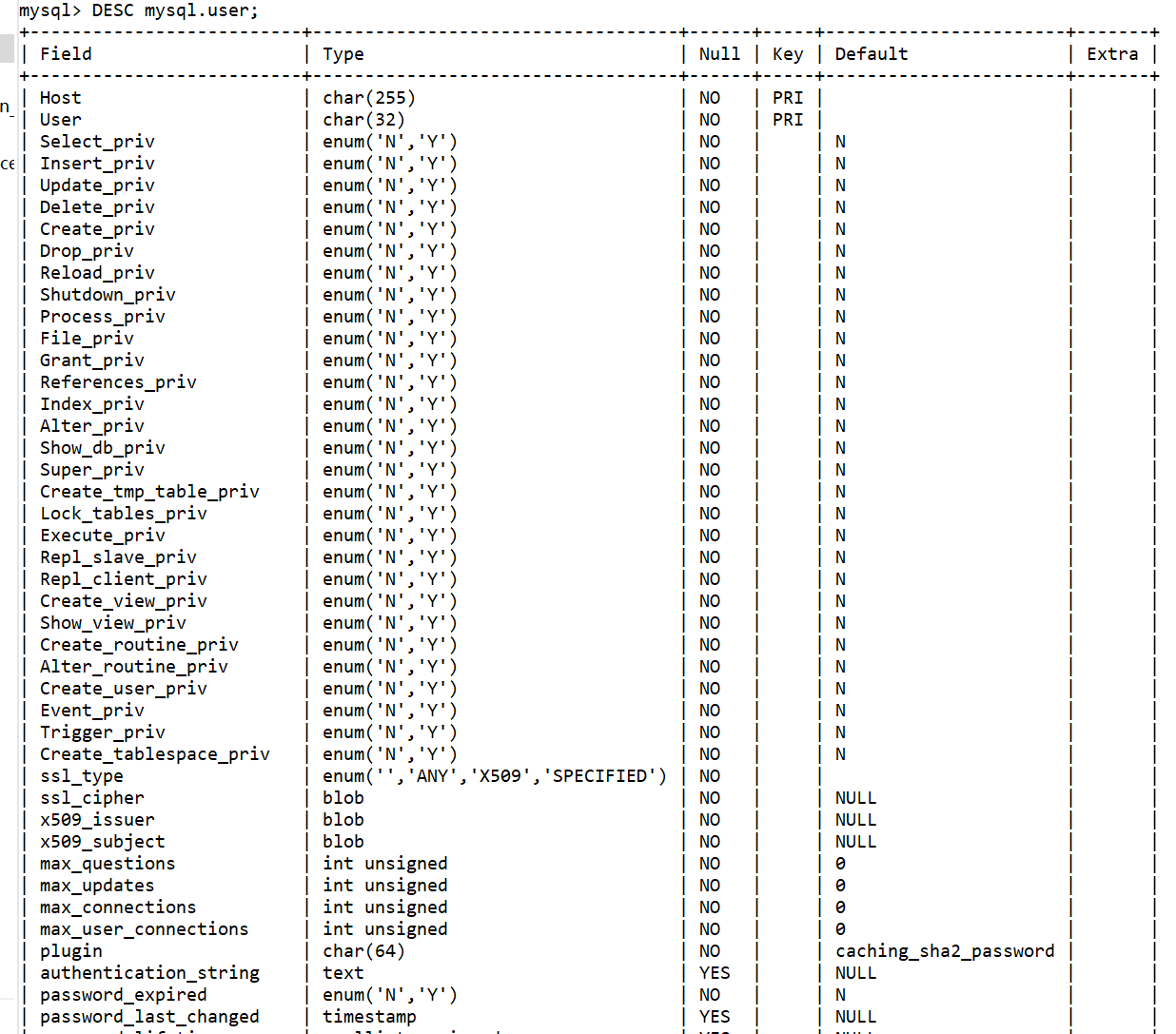
方式一：通过查询 mysql 数据库的 user 表

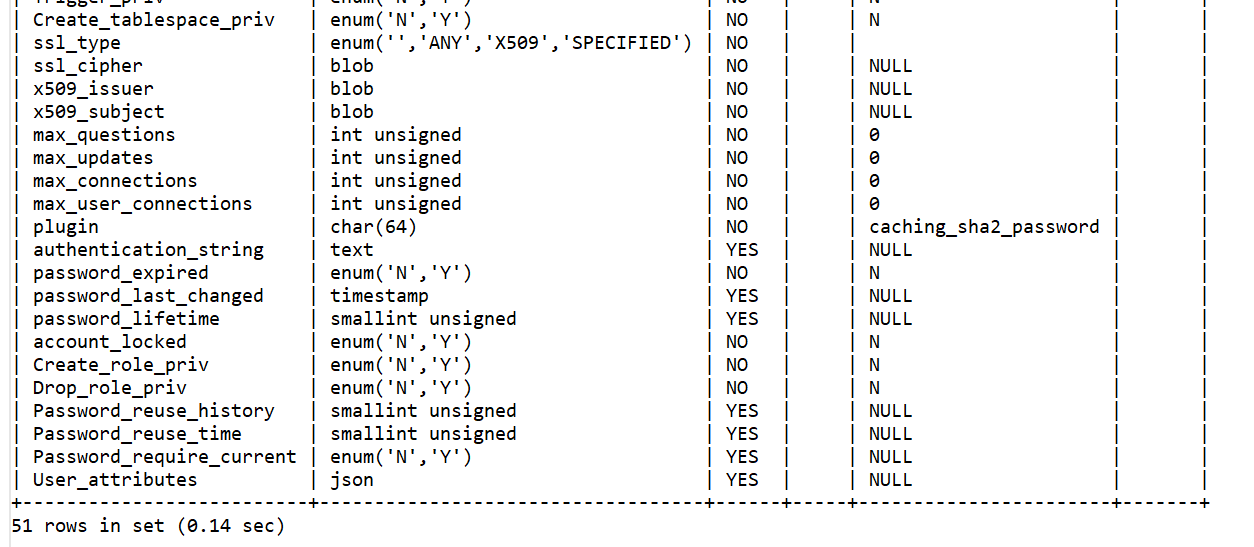
Mysql8.4 将把所有用户的信息放在系统自带的 mysql 数据库的 user 表中，所以查询该表能够得到用户及其权限信息。

(1)查看 user 表的结构

命令： root>DESC mysql.user;

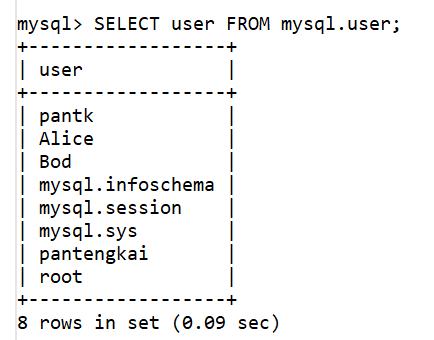
user 表有 51 个字段，所有以 priv 结尾的字段名均为用户是否具有该字段对应的权限（值为‘Y’或‘N’）



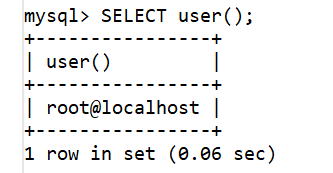


(2)查看用户列表

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

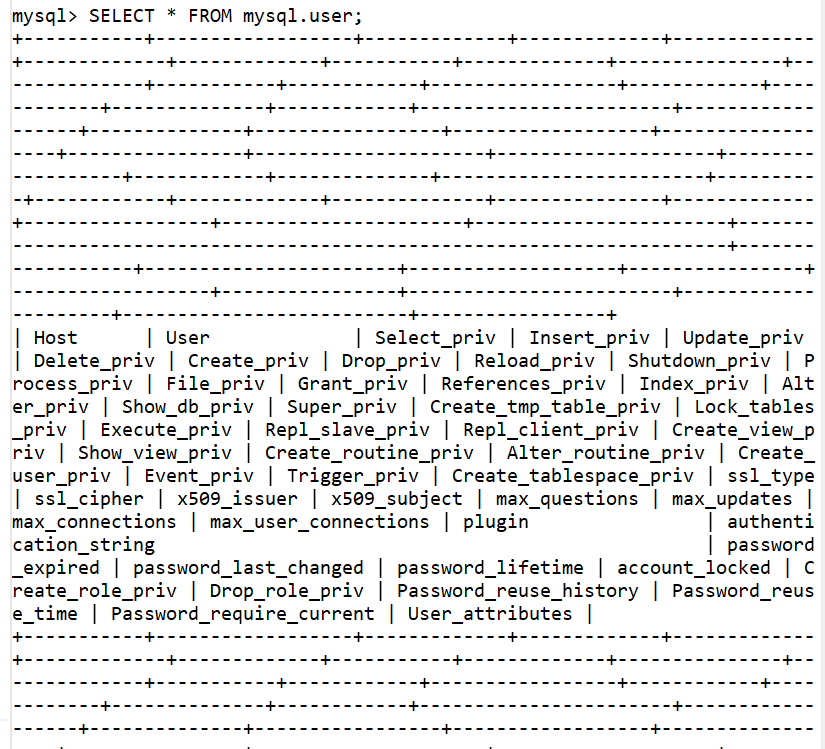


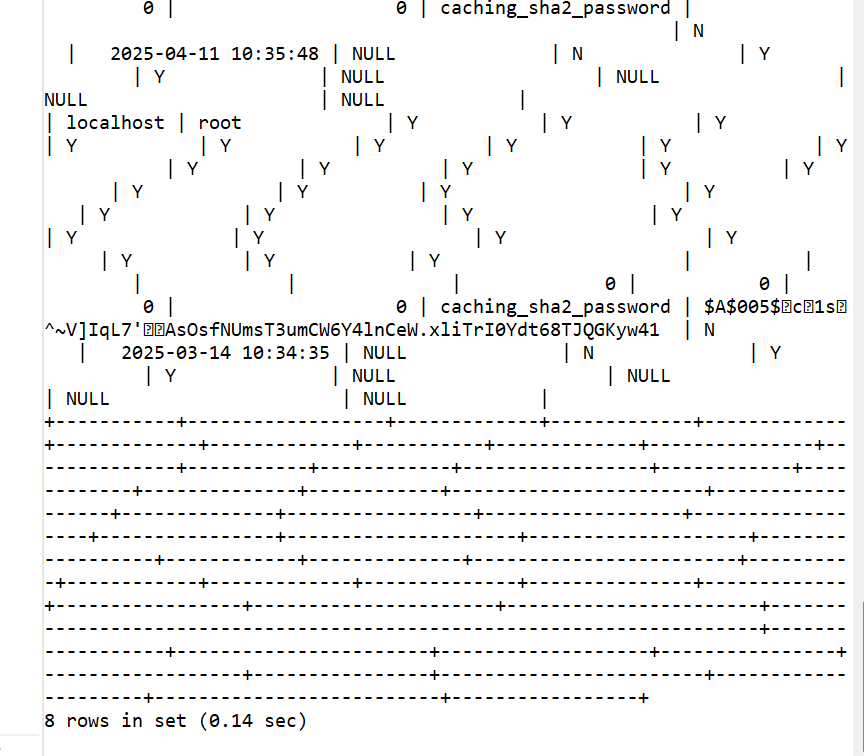
 使用命令 SELECT user(); 可以查询当前用户名，这对不记得当前用户是谁非常有用



(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

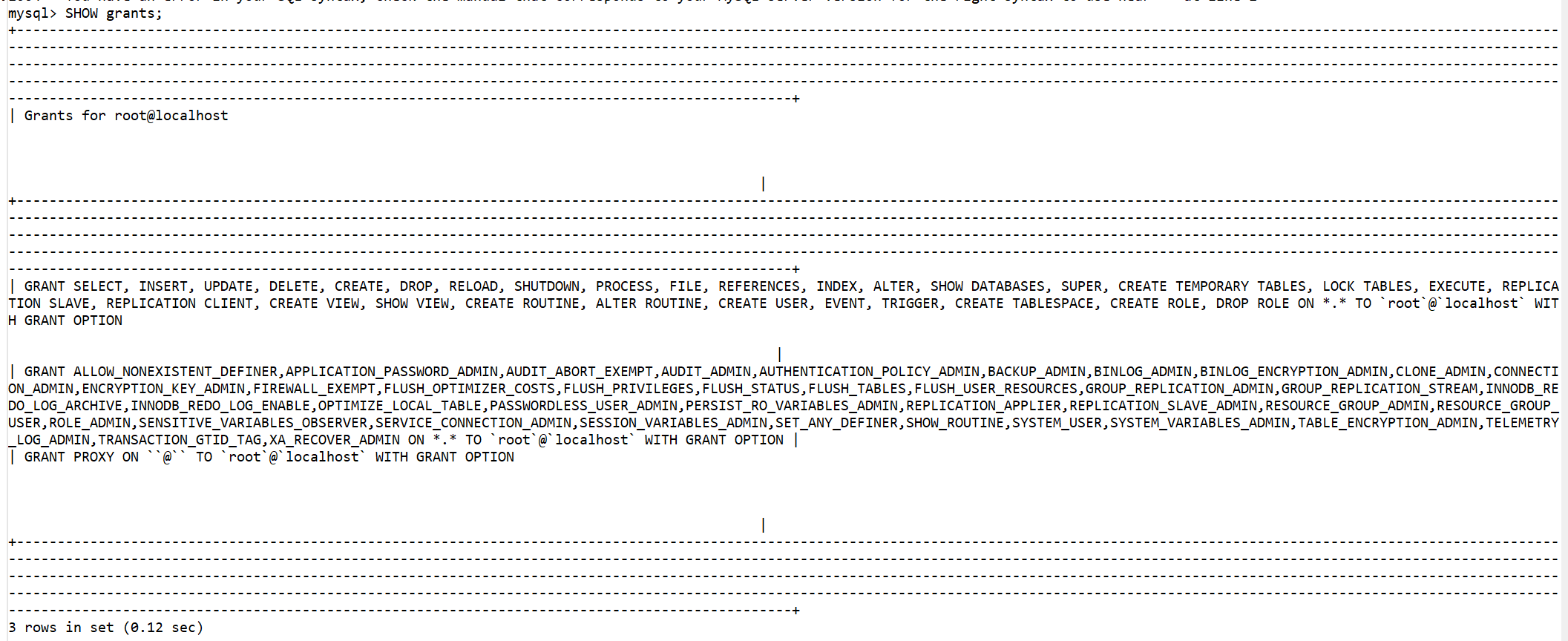
可以使用 SHOW 命令显示当前用户或特定用户的权限

(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：

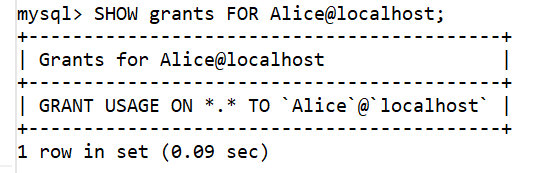
SHOW grants;

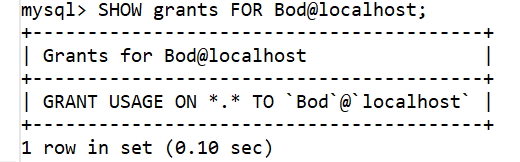
SHOW grants for current\_user; Show grants for current\_user();



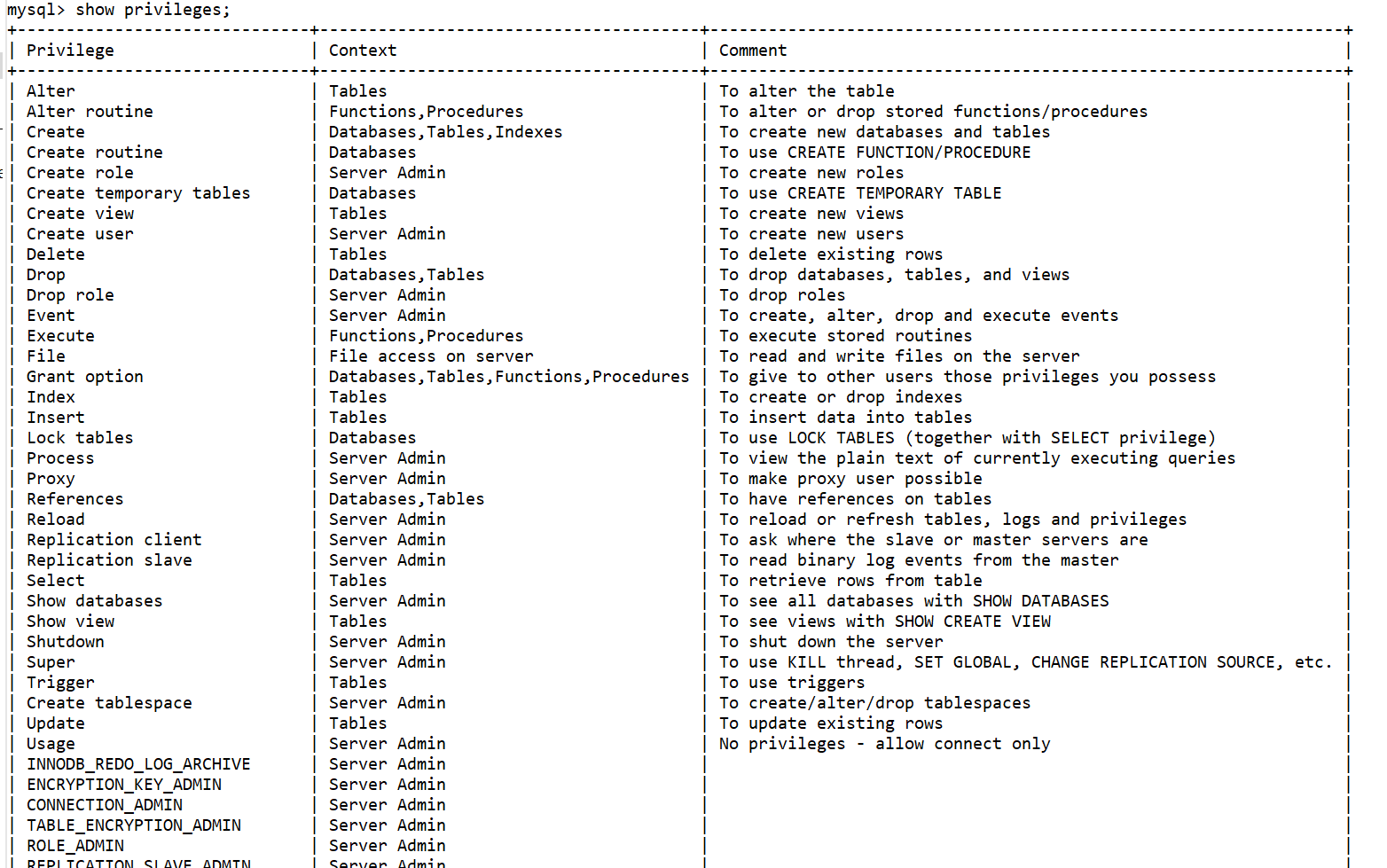
(6)查询特定用户的权限

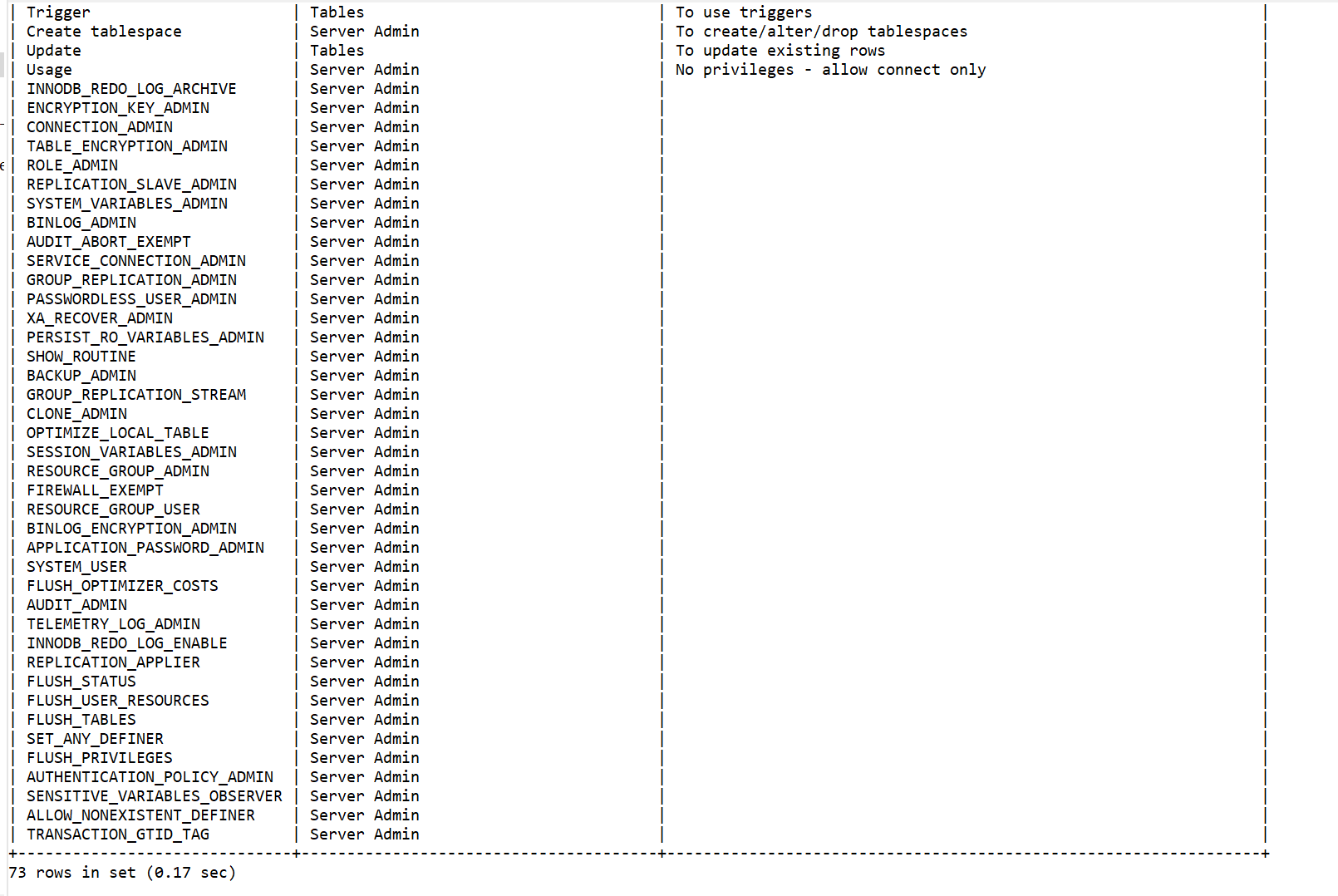
命令： root>show grants for user\_name@localhost;





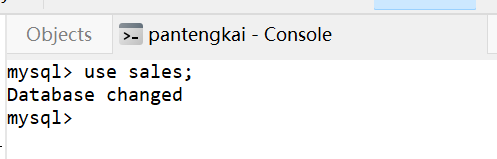
3.查看系统的所有权限



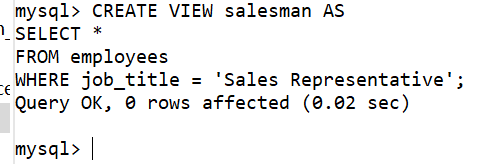


4.创建视图

(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

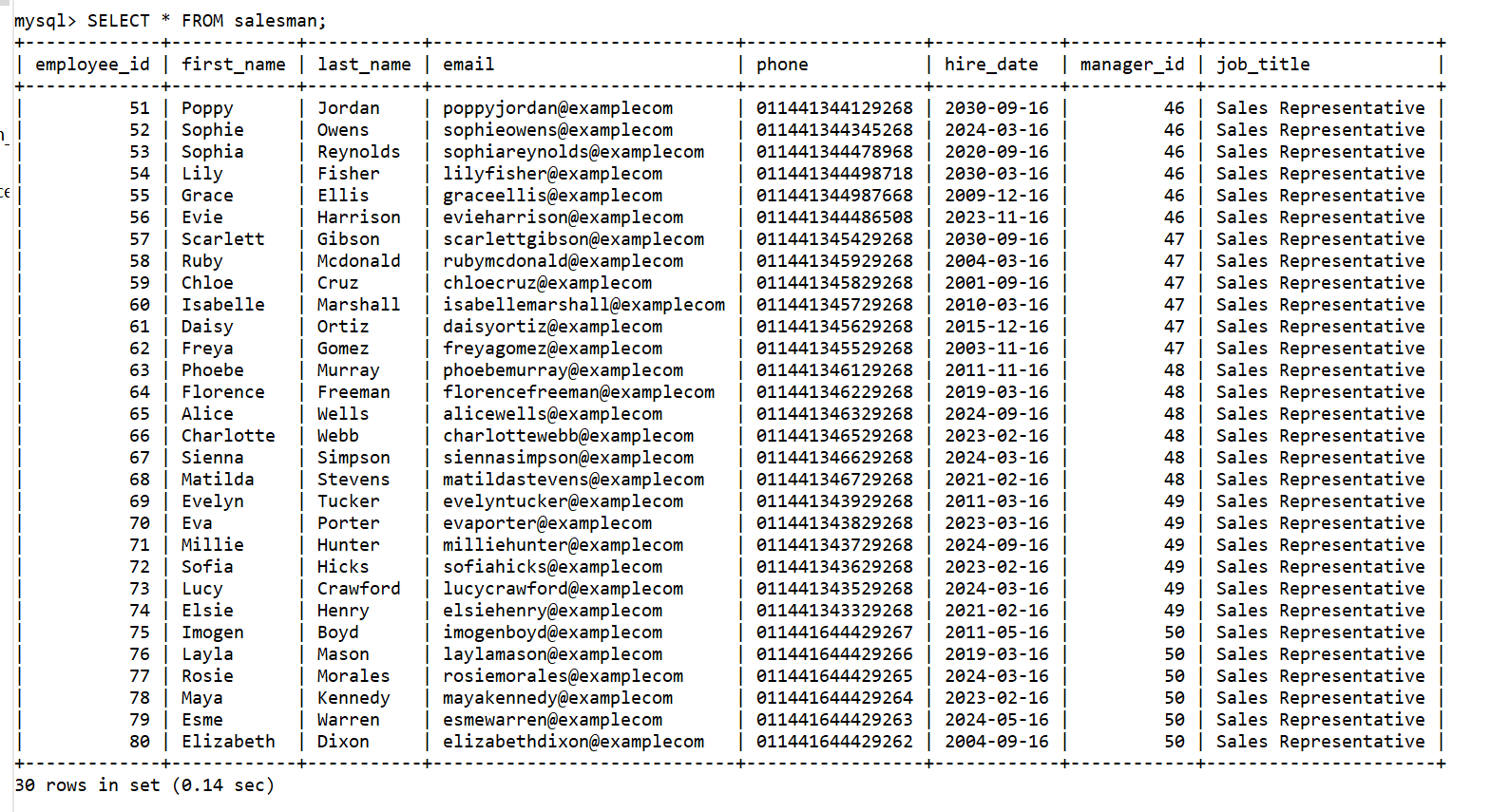


1. 创建视图 salesman，该视图只保存 employees 表中所有 job\_title 为'Sales Representative'的雇员



1. 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

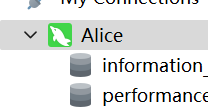


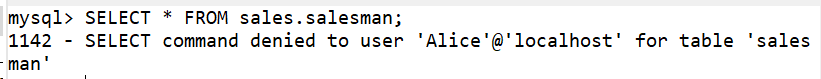
5.用户授权与权限回收

(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;

CMD密码忘了，用Navicat

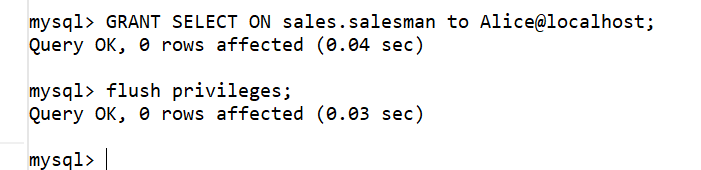




观察结果

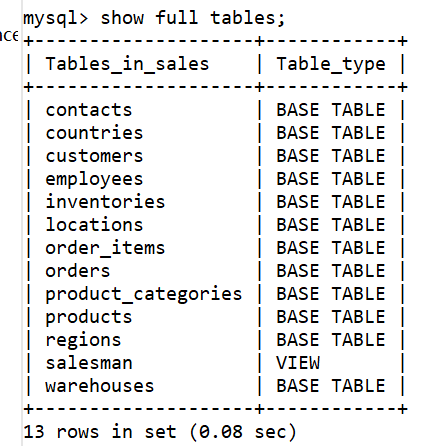
(2)用户 user1 将查看 sales.salesman 的权限授予给 Alice，命令：

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



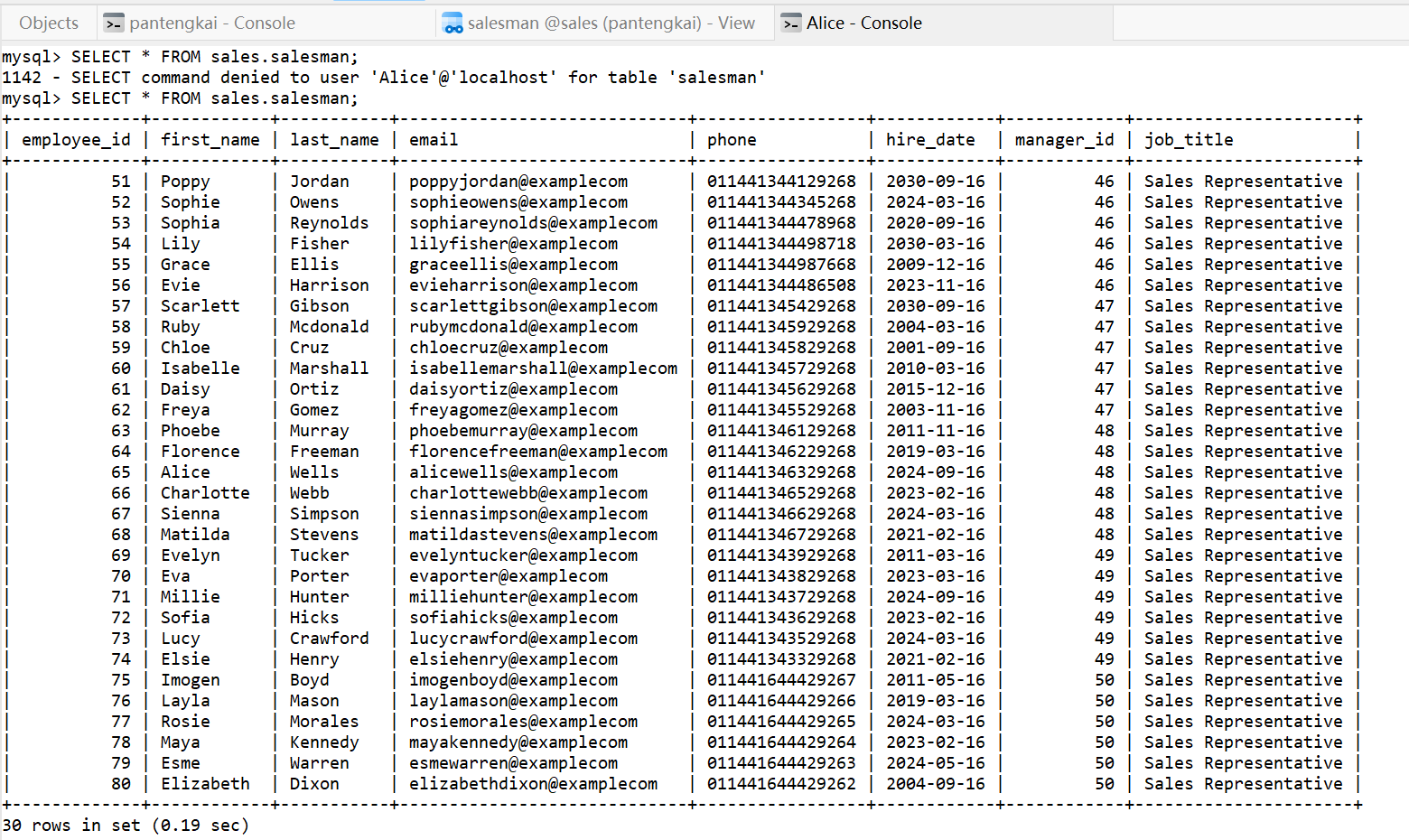
(3) 用户 user1 发布语句： user1>show full tables;

--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;

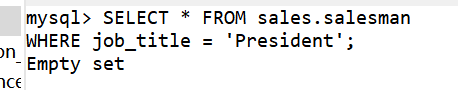
观察结果并与第(1)步结果比较，理解差异背后的原因



这次查询就有结果了

**(5)**Alice 发布语句： Alice>SELECT \* FROM sales.salesman where job\_title=’President’;

试解释该结果

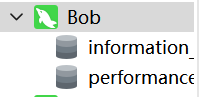


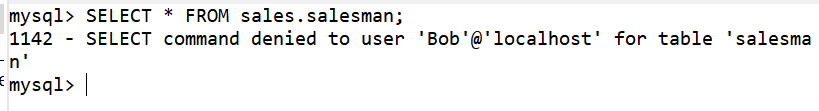
因为该视图中所有job\_title都是sales representative，所以查询President的都为空集

(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

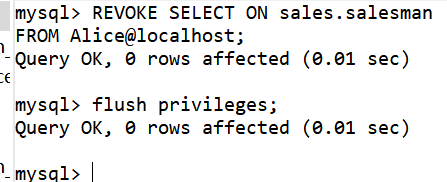




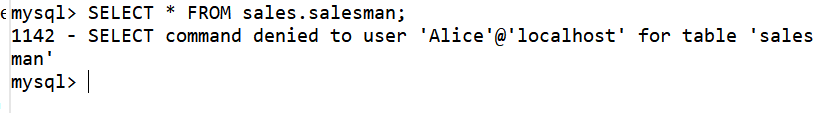
无权限

(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;



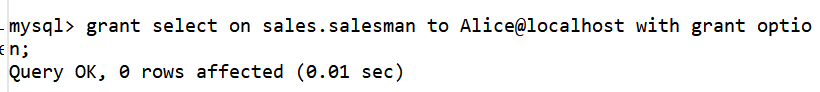
1. 用户 Alice 重新发布语句： Alice>SELECT \* FROM sales.salesman; 观察结果



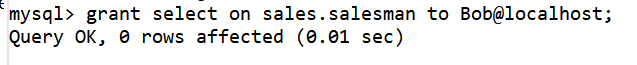
由于权限被收回查询失败

(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;



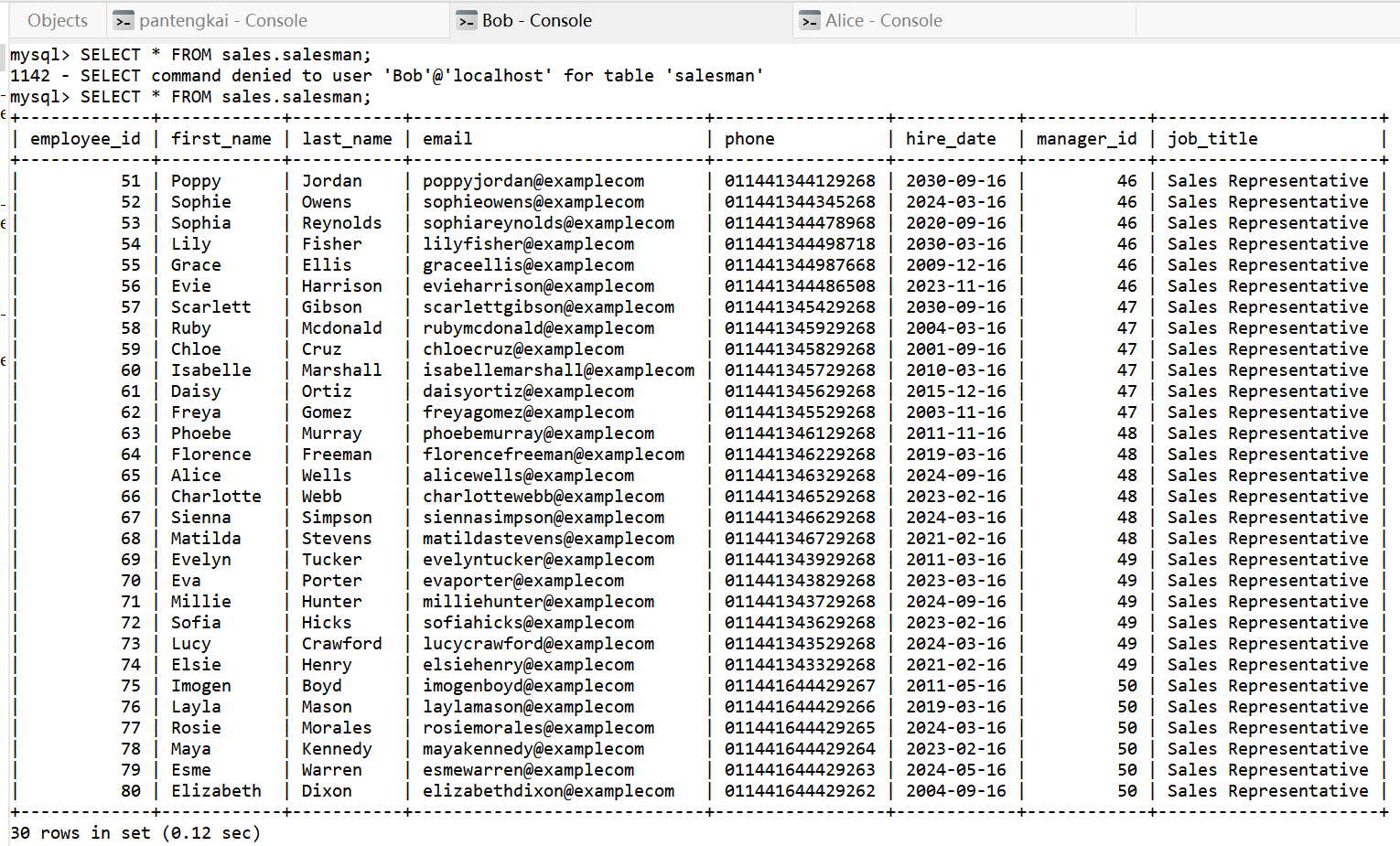
(10)用户 Alice 发布授权给 Bob 语句： Alice>grant select on sales.salesman to Bob@localhost;



(11)用户 Bob 发布查询语句：

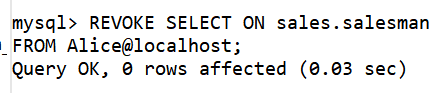
Bob>SELECT \* FROM sales.salesman;

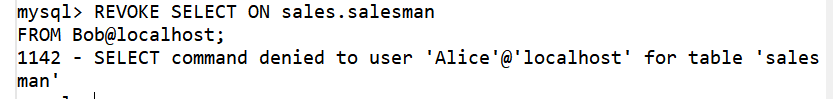
观察系统反馈结果



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）

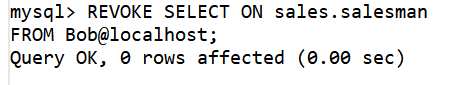
(12)user1 回收授予给 Alice 的权限， Alice 回收授予给 Bob 的权限（这步为角色操作做准备）

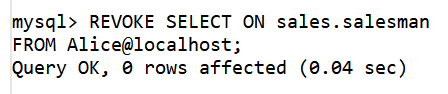




Alice回收权限失败

换一下顺序：





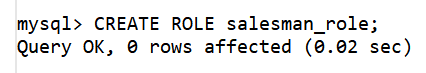
只能先删低级的再删高级的

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

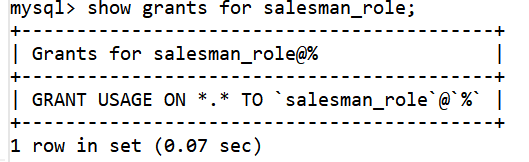
user1> create role salesman\_role;

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



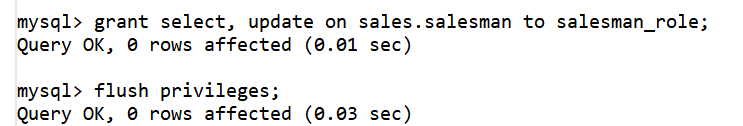
(2)查看 salesman\_role 角色包含的权限

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



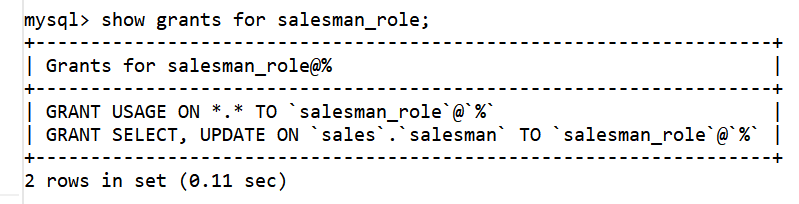
(3)将 select 和 update 视图 Salesman 的权限授予给 salesman\_role 角色

user1> grant select, 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

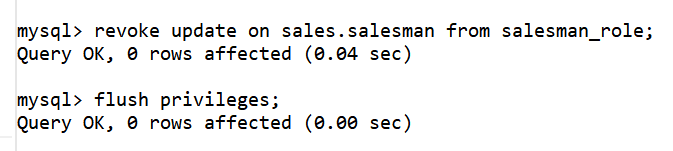
user1> show grants for salesman\_role;



多了查询和更新salesman视图的权限

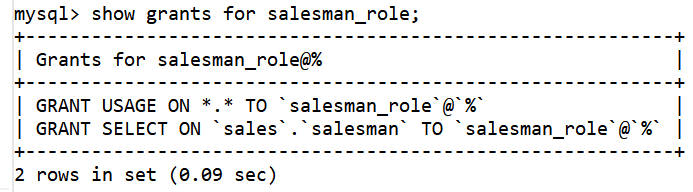
(5)回收 salesman\_role 角色中的 update 权限

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;

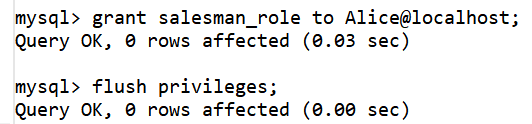


少了更新权限

(7)将角色授权给用户

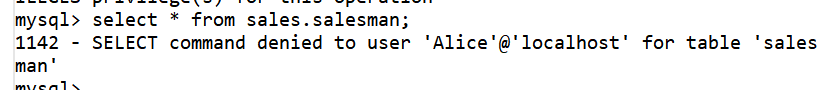
user1> grant salesman\_role to Alice@localhost; user1> flush privileges;

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

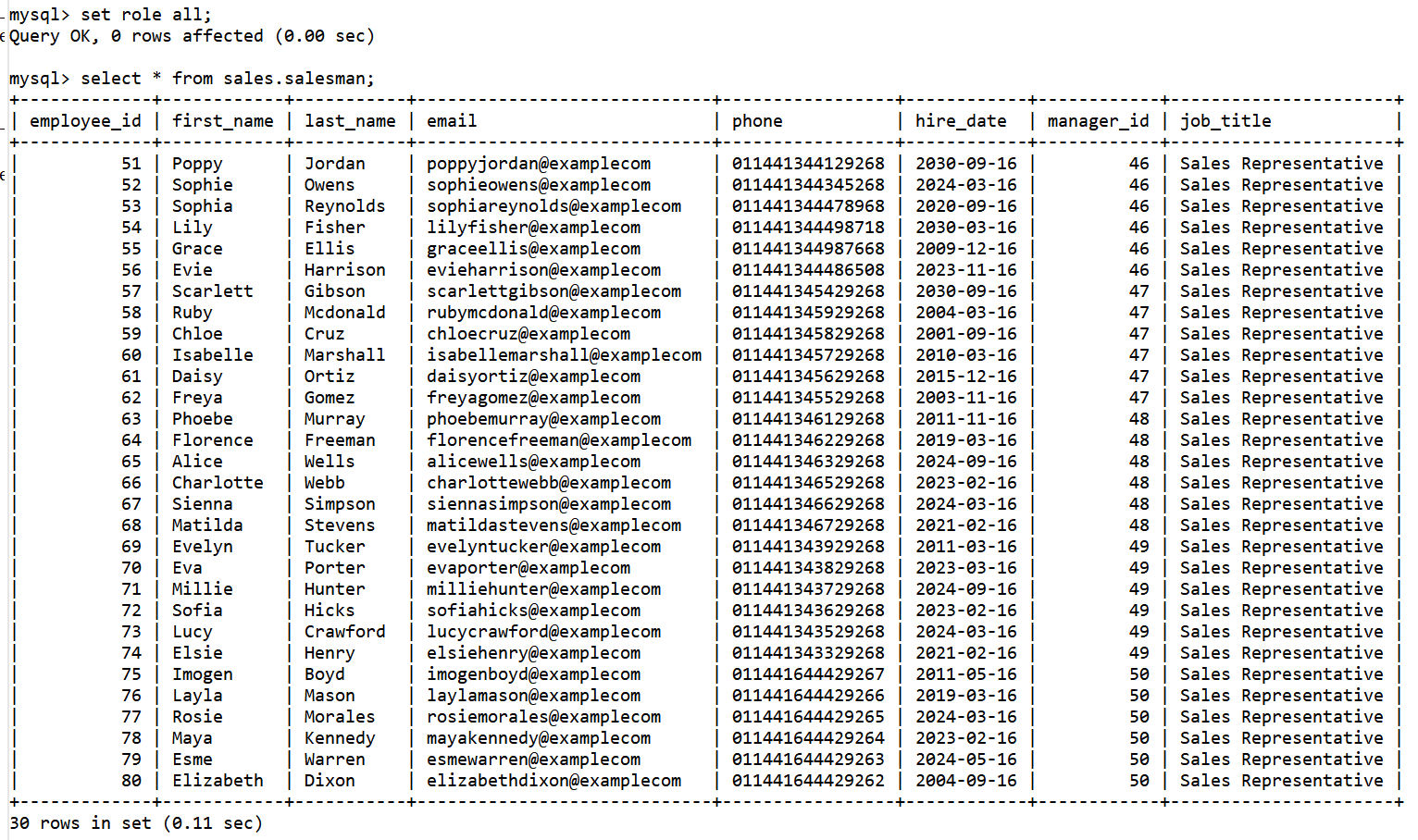
Alice> select \* from sales.salesman;



系统可能提示以下错误：

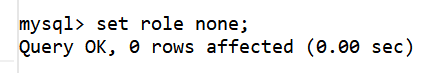
ERROR 1142 (42000): SELECT command denied to user 'Alice'@'localhost' for table 'salesman'原因： mysql 中角色使用先设定是否启用，本实验仅介绍会话级角色的启用与关闭Alice> set role salesman\_role; 或者 Alice> set role all; Alice> select \* from sales.salesman;

角色启用，生效



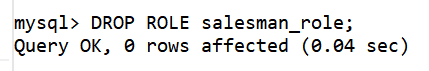
关闭当前会话的生效角色

Alice> set role none;



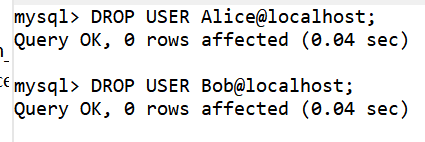
(9)删除角色

user1> drop role salesman\_role;



(10)删除用户

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



1. 实验总结

3.1 完成的工作

完成创造、删除角色或用户操作，完成给予、查询、收回权限等操作。

3.2 对实验的认识

思考题：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？

答：（1）主体名称（必需）：由字母、数字、下划线（\_）、美元符号（$）和点号（.）组成，但 不能以数字开头。

主机名限定符（可选）：使用 @ 符号分隔主体名称和主机名，用于限制该用户 / 角色只能从特定主机连接。

（2）要注意大小写敏感性：用户名和角色名在 Windows 上不区分大小写，在 Linux/macOS 上默认区分大小写。

特殊字符处理：若名称包含特殊字符（如 @、.），需用引号分隔各部分。例如：'user.name'@'example.com' 或 `user@domain`@`192.168.1.1`。

权限范围：角色仅用于组织权限，用户需显式激活角色才能使用其权限。

安全性：避免使用通配符主机（'%'），尽量限制用户连接来源。定期清理不再使用的用户和角色。

1. 步骤 5 之(5)的结果与视图有关吗？试解释之。

答：有关。视图 salesman 里的数据本身就仅包含 job\_title 为 'Sales Representative' 的记录，不存在 job\_title 是 'President' 的数据 。

数据库对视图的查询，实际是在视图定义的基础数据集上进行筛选。此次查询条件要求找 job\_title 为 'President' 的记录，而视图中根本没有这类数据，所以最终返回 Empty set，即空集 。

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。

答：不能级联回收，如本实验中Alice将权限转授给Bob，即使用CASCADE回收Alice的权限，Bob仍然保有Alice给它的权限。

3.3 遇到的困难及解决方法

无。