



## 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

## 2. 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

```
mysql> SELECT user FROM mysql.user;
```

user
pantk
mysql.infoschema
mysql.session
mysql.sys
pantengkai
root

```
6 rows in set (0.07 sec)
```

(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
```

```
mysql> CREATE USER Alice@localhost IDENTIFIED BY 'root';
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE USER Bod@localhost IDENTIFIED BY 'root';
Query OK, 0 rows affected (0.04 sec)
```

## 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' ）

```
mysql> DESC mysql.user;
```

	Field	Type	Null	Key	Default	Extra
n	Host	char(255)	NO	PRI		
	User	char(32)	NO	PRI		
ce	Select_priv	enum('N','Y')	NO		N	
	Insert_priv	enum('N','Y')	NO		N	
	Update_priv	enum('N','Y')	NO		N	
	Delete_priv	enum('N','Y')	NO		N	
	Create_priv	enum('N','Y')	NO		N	
	Drop_priv	enum('N','Y')	NO		N	
	Reload_priv	enum('N','Y')	NO		N	
	Shutdown_priv	enum('N','Y')	NO		N	
	Process_priv	enum('N','Y')	NO		N	
	File_priv	enum('N','Y')	NO		N	
	Grant_priv	enum('N','Y')	NO		N	
	References_priv	enum('N','Y')	NO		N	
	Index_priv	enum('N','Y')	NO		N	
	Alter_priv	enum('N','Y')	NO		N	
	Show_db_priv	enum('N','Y')	NO		N	
	Super_priv	enum('N','Y')	NO		N	
	Create_tmp_table_priv	enum('N','Y')	NO		N	
	Lock_tables_priv	enum('N','Y')	NO		N	
	Execute_priv	enum('N','Y')	NO		N	
	Repl_slave_priv	enum('N','Y')	NO		N	
	Repl_client_priv	enum('N','Y')	NO		N	
	Create_view_priv	enum('N','Y')	NO		N	
	Show_view_priv	enum('N','Y')	NO		N	
	Create_routine_priv	enum('N','Y')	NO		N	
	Alter_routine_priv	enum('N','Y')	NO		N	
	Create_user_priv	enum('N','Y')	NO		N	
	Event_priv	enum('N','Y')	NO		N	
	Trigger_priv	enum('N','Y')	NO		N	
	Create_tablespace_priv	enum('N','Y')	NO		N	
	ssl_type	enum('', 'ANY', 'X509', 'SPECIFIED')	NO			
	ssl_cipher	blob	NO		NULL	
	x509_issuer	blob	NO		NULL	
	x509_subject	blob	NO		NULL	
	max_questions	int unsigned	NO		0	
	max_updates	int unsigned	NO		0	
	max_connections	int unsigned	NO		0	
	max_user_connections	int unsigned	NO		0	
	plugin	char(64)	NO		caching_sha2_password	
	authentication_string	text	YES		NULL	
	password_expired	enum('N','Y')	NO		N	
	password_last_changed	timestamp	YES		NULL	
	Create_tablespace_priv	enum('N','Y')	NO		N	
	ssl_type	enum('', 'ANY', 'X509', 'SPECIFIED')	NO			
	ssl_cipher	blob	NO		NULL	
	x509_issuer	blob	NO		NULL	
	x509_subject	blob	NO		NULL	
	max_questions	int unsigned	NO		0	
	max_updates	int unsigned	NO		0	
	max_connections	int unsigned	NO		0	
	max_user_connections	int unsigned	NO		0	
	plugin	char(64)	NO		caching_sha2_password	
	authentication_string	text	YES		NULL	
	password_expired	enum('N','Y')	NO		N	
	password_last_changed	timestamp	YES		NULL	
	password_lifetime	smallint unsigned	YES		NULL	
	account_locked	enum('N','Y')	NO		N	
	Create_role_priv	enum('N','Y')	NO		N	
	Drop_role_priv	enum('N','Y')	NO		N	
	Password_reuse_history	smallint unsigned	YES		NULL	
	Password_reuse_time	smallint unsigned	YES		NULL	
	Password_require_current	enum('N','Y')	YES		NULL	
	User_attributes	json	YES		NULL	

51 rows in set (0.14 sec)

## (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

```
mysql> SELECT user FROM mysql.user;
```

user
pantk
Alice
Bod
mysql.infoschema
mysql.session
mysql.sys
pantengkai
root

```
8 rows in set (0.09 sec)
```

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

```
mysql> SELECT user();
```

user()
root@localhost

```
1 row in set (0.06 sec)
```

(3)查询所有用户的权限

命令： `root>SELECT * FROM mysql.user;`

```
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
| Host          | User                | Select_priv | Insert_priv | Update_priv |
| Delete_priv   | Create_priv         | Drop_priv   | Reload_priv | Shutdown_priv | P
rocess_priv    | File_priv          | Grant_priv  | References_priv | Index_priv | Alt
er_priv       | Show_db_priv      | Super_priv  | Create_tmp_table_priv | Lock_tables
_priv        | Execute_priv     | Repl_slave_priv | Repl_client_priv | Create_view_p
riv          | Show_view_priv   | Create_routine_priv | Alter_routine_priv | Create_
user_priv   | Event_priv      | Trigger_priv | Create_tablespace_priv | ssl_type
| ssl_cipher   | x509_issuer      | x509_subject | max_questions | max_updates |
max_connections | max_user_connections | plugin              | authenti
cation_string | password
_expired | password_last_changed | password_lifetime | account_locked | C
reate_role_priv | Drop_role_priv | Password_reuse_history | Password_reus
e_time | Password_require_current | User_attributes |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
```





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();

```
mysql> SHOW grants;
+-----+
| Grants for root@localhost |
+-----+
| GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, SHUTDOWN, PROCESS, FILE, REFERENCES, INDEX, ALTER, SHOW DATABASES, SUPER, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, CREATE TABLESPACE, CREATE ROLE, DROP ROLE ON *.* TO 'root'@'localhost' WITH GRANT OPTION |
+-----+
| GRANT ALLOW_NONEXISTENT_DEFINER, APPLICATION_PASSWORD_ADMIN, AUDIT_ABORT_EXEMPT, AUDIT_ADMIN, AUTHENTICATION_POLICY_ADMIN, BACKUP_ADMIN, BINLOG_ADMIN, BINLOG_ENCRYPTION_ADMIN, CLONE_ADMIN, CONNECTION_ADMIN, ENCRYPTION_KEY_ADMIN, FIREWALL_EXEMPT, FLUSH_OPTIMIZER_COSTS, FLUSH_PRIVILEGES, FLUSH_STATUS, FLUSH_TABLES, FLUSH_USER_RESOURCES, GROUP_REPLICATION_ADMIN, GROUP_REPLICATION_STREAM, INNODB_REDO_LOG_ARCHIVE, INNODB_REDO_LOG_ENABLE, OPTIMIZE_LOCAL_TABLE, PASSWORDLESS_USER_ADMIN, PERSIST_RO_VARIABLES_ADMIN, REPLICATION_APPLIER, REPLICATION_SLAVE_ADMIN, RESOURCE_GROUP_ADMIN, RESOURCE_GROUP_USER, ROLE_ADMIN, SENSITIVE_VARIABLES_OBSERVER, SERVICE_CONNECTION_ADMIN, SESSION_VARIABLES_ADMIN, SET_ANY_DEFINER, SHOW_ROUTINE, SYSTEM_USER, SYSTEM_VARIABLES_ADMIN, TABLE_ENCRYPTION_ADMIN, TELEMETRY_LOG_ADMIN, TRANSACTION_GTID_TAG, XA_RECOVER_ADMIN ON *.* TO 'root'@'localhost' WITH GRANT OPTION |
+-----+
| GRANT PROXY ON *.* TO 'root'@'localhost' WITH GRANT OPTION |
+-----+
3 rows in set (0.12 sec)
```

## (6)查询特定用户的权限

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

```
mysql> SHOW grants FOR Alice@localhost;
+-----+
| Grants for Alice@localhost |
+-----+
| GRANT USAGE ON *.* TO `Alice`@`localhost` |
+-----+
1 row in set (0.09 sec)
```

```
mysql> SHOW grants FOR Bod@localhost;
+-----+
| Grants for Bod@localhost |
+-----+
| GRANT USAGE ON *.* TO `Bod`@`localhost` |
+-----+
1 row in set (0.10 sec)
```

## 3.查看系统的所有权限



```
mysql> show privileges;
```

Privilege	Context	Comment
Alter	Tables	To alter the table
Alter routine	Functions,Procedures	To alter or drop stored functions/procedures
Create	Databases,Tables,Indexes	To create new databases and tables
Create routine	Databases	To use CREATE FUNCTION/PROCEDURE
Create role	Server Admin	To create new roles
Create temporary tables	Databases	To use CREATE TEMPORARY TABLE
Create view	Tables	To create new views
Create user	Server Admin	To create new users
Delete	Tables	To delete existing rows
Drop	Databases,Tables	To drop databases, tables, and views
Drop role	Server Admin	To drop roles
Event	Server Admin	To create, alter, drop and execute events
Execute	Functions,Procedures	To execute stored routines
File	File access on server	To read and write files on the server
Grant option	Databases,Tables,Functions,Procedures	To give to other users those privileges you possess
Index	Tables	To create or drop indexes
Insert	Tables	To insert data into tables
Lock tables	Databases	To use LOCK TABLES (together with SELECT privilege)
Process	Server Admin	To view the plain text of currently executing queries
Proxy	Server Admin	To make proxy user possible
References	Databases,Tables	To have references on tables
Reload	Server Admin	To reload or refresh tables, logs and privileges
Replication client	Server Admin	To ask where the slave or master servers are
Replication slave	Server Admin	To read binary log events from the master
Select	Tables	To retrieve rows from table
Show databases	Server Admin	To see all databases with SHOW DATABASES
Show view	Tables	To see views with SHOW CREATE VIEW
Shutdown	Server Admin	To shut down the server
Super	Server Admin	To use KILL thread, SET GLOBAL, CHANGE REPLICATION SOURCE, etc.
Trigger	Tables	To use triggers
Create tablespace	Server Admin	To create/alter/drop tablespaces
Update	Tables	To update existing rows
Usage	Server Admin	No privileges - allow connect only
INNODB_REDO_LOG_ARCHIVE	Server Admin	
ENCRYPTION_KEY_ADMIN	Server Admin	
CONNECTION_ADMIN	Server Admin	
TABLE_ENCRYPTION_ADMIN	Server Admin	
ROLE_ADMIN	Server Admin	
REPLICATION_SLAVE_ADMIN	Server Admin	
SYSTEM_VARIABLES_ADMIN	Server Admin	
BINLOG_ADMIN	Server Admin	
AUDIT_ABORT_EXEMPT	Server Admin	
SERVICE_CONNECTION_ADMIN	Server Admin	
GROUP_REPLICATION_ADMIN	Server Admin	
PASSWORDLESS_USER_ADMIN	Server Admin	
XA_RECOVER_ADMIN	Server Admin	
PERSIST_RO_VARIABLES_ADMIN	Server Admin	
SHOW_ROUTINE	Server Admin	
BACKUP_ADMIN	Server Admin	
GROUP_REPLICATION_STREAM	Server Admin	
CLONE_ADMIN	Server Admin	
OPTIMIZE_LOCAL_TABLE	Server Admin	
SESSION_VARIABLES_ADMIN	Server Admin	
RESOURCE_GROUP_ADMIN	Server Admin	
FIREWALL_EXEMPT	Server Admin	
RESOURCE_GROUP_USER	Server Admin	
BINLOG_ENCRYPTION_ADMIN	Server Admin	
APPLICATION_PASSWORD_ADMIN	Server Admin	
SYSTEM_USER	Server Admin	
FLUSH_OPTIMIZER_COSTS	Server Admin	
AUDIT_ADMIN	Server Admin	
TELEMETRY_LOG_ADMIN	Server Admin	
INNODB_REDO_LOG_ENABLE	Server Admin	
REPLICATION_APPLIER	Server Admin	
FLUSH_STATUS	Server Admin	
FLUSH_USER_RESOURCES	Server Admin	
FLUSH_TABLES	Server Admin	
SET_ANY_DEFINER	Server Admin	
FLUSH_PRIVILEGES	Server Admin	
AUTHENTICATION_POLICY_ADMIN	Server Admin	
SENSITIVE_VARIABLES_OBSERVER	Server Admin	
ALLOW_NONEXISTENT_DEFINER	Server Admin	
TRANSACTION_GTID_TAG	Server Admin	

73 rows in set (0.17 sec)

## 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

```

Objects  >- pantengkai - Console

mysql> use sales;
Database changed
mysql>

```

(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

```

mysql> CREATE VIEW salesman AS
SELECT *
FROM employees
WHERE job_title = 'Sales Representative';
Query OK, 0 rows affected (0.02 sec)

mysql> |

```

(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

```

mysql> SELECT * FROM salesman;

```

employee_id	first_name	last_name	email	phone	hire_date	manager_id	job_title
51	Poppy	Jordan	poppyjordan@examplecom	011441344129268	2030-09-16	46	Sales Representative
52	Sophie	Owens	sophieowens@examplecom	011441344345268	2024-03-16	46	Sales Representative
53	Sophia	Reynolds	sophiareynolds@examplecom	011441344478968	2020-09-16	46	Sales Representative
54	Lily	Fisher	lilyfisher@examplecom	011441344498718	2030-03-16	46	Sales Representative
55	Grace	Ellis	graceellis@examplecom	011441344987668	2009-12-16	46	Sales Representative
56	Evie	Harrison	evieharrison@examplecom	011441344486508	2023-11-16	46	Sales Representative
57	Scarlett	Gibson	scarlettgibson@examplecom	011441345429268	2030-09-16	47	Sales Representative
58	Ruby	Mcdonald	rubymcdonald@examplecom	011441345929268	2004-03-16	47	Sales Representative
59	Chloe	Cruz	chloecruz@examplecom	011441345829268	2001-09-16	47	Sales Representative
60	Isabelle	Marshall	isabellemarshall@examplecom	011441345729268	2010-03-16	47	Sales Representative
61	Daisy	Ortiz	daisyortiz@examplecom	011441345629268	2015-12-16	47	Sales Representative
62	Freya	Gomez	freitagomez@examplecom	011441345529268	2003-11-16	47	Sales Representative
63	Phoebe	Murray	phoebemurray@examplecom	011441346129268	2011-11-16	48	Sales Representative
64	Florence	Freeman	florencefreeman@examplecom	011441346229268	2019-03-16	48	Sales Representative
65	Alice	Wells	alicewells@examplecom	011441346329268	2024-09-16	48	Sales Representative
66	Charlotte	Webb	charlottewebb@examplecom	011441346529268	2023-02-16	48	Sales Representative
67	Sienna	Simpson	siennasimpson@examplecom	011441346629268	2024-03-16	48	Sales Representative
68	Matilda	Stevens	matildastevens@examplecom	011441346729268	2021-02-16	48	Sales Representative
69	Evelyn	Tucker	evelyntucker@examplecom	011441343929268	2011-03-16	49	Sales Representative
70	Eva	Porter	evaporter@examplecom	011441343829268	2023-03-16	49	Sales Representative
71	Millie	Hunter	milliehunter@examplecom	011441343729268	2024-09-16	49	Sales Representative
72	Sofia	Hicks	sofiiahicks@examplecom	011441343629268	2023-02-16	49	Sales Representative
73	Lucy	Crawford	lucycrawford@examplecom	011441343529268	2024-03-16	49	Sales Representative
74	Elsie	Henry	elsiehenry@examplecom	011441343329268	2021-02-16	49	Sales Representative
75	Imogen	Boyd	imogenboyd@examplecom	011441644429267	2011-05-16	50	Sales Representative
76	Layla	Mason	laylamason@examplecom	011441644429266	2019-03-16	50	Sales Representative
77	Rosie	Morales	rosiemorales@examplecom	011441644429265	2024-03-16	50	Sales Representative
78	Maya	Kennedy	mayakennedy@examplecom	011441644429264	2023-02-16	50	Sales Representative
79	Esme	Warren	esmewarren@examplecom	011441644429263	2024-05-16	50	Sales Representative
80	Elizabeth	Dixon	elizabethdixon@examplecom	011441644429262	2004-09-16	50	Sales Representative

```

30 rows in set (0.14 sec)

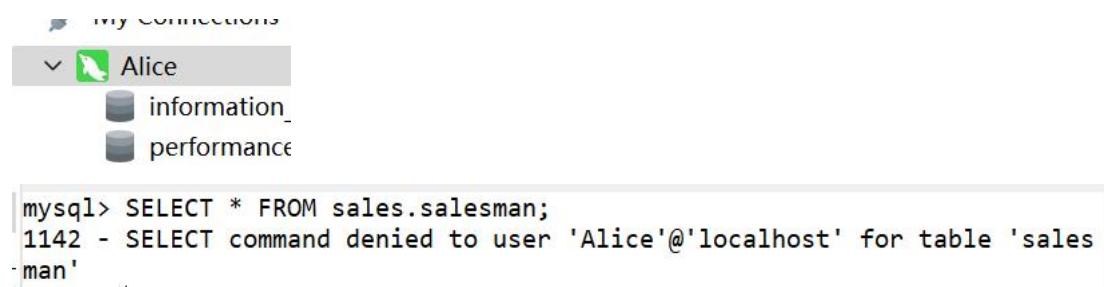
```

## 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

```
mysql> GRANT SELECT ON sales.salesman to Alice@localhost;
Query OK, 0 rows affected (0.04 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.03 sec)

mysql> |
```

(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

```
mysql> show full tables;
+-----+-----+
| Tables_in_sales | Table_type |
+-----+-----+
| contacts        | BASE TABLE |
| countries       | BASE TABLE |
| customers       | BASE TABLE |
| employees       | BASE TABLE |
| inventories     | BASE TABLE |
| locations       | BASE TABLE |
| order_items     | BASE TABLE |
| orders         | BASE TABLE |
| product_categories | BASE TABLE |
| products       | BASE TABLE |
| regions        | BASE TABLE |
| salesman        | VIEW       |
| warehouses     | BASE TABLE |
+-----+-----+
13 rows in set (0.08 sec)
```



(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

Objects | pantengkai - Console | salesman @sales (pantengkai) - View | Alice - Console

```
mysql> SELECT * FROM sales.salesman;
1142 - SELECT command denied to user 'Alice'@'localhost' for table 'salesman'
mysql> SELECT * FROM sales.salesman;
```

employee_id	first_name	last_name	email	phone	hire_date	manager_id	job_title
51	Poppy	Jordan	poppyjordan@examplecom	011441344129268	2030-09-16	46	Sales Representative
52	Sophie	Owens	sophieowens@examplecom	011441344345268	2024-03-16	46	Sales Representative
53	Sophia	Reynolds	sophiareynolds@examplecom	011441344478968	2020-09-16	46	Sales Representative
54	Lily	Fisher	lilyfisher@examplecom	011441344498718	2030-03-16	46	Sales Representative
55	Grace	Ellis	graceellis@examplecom	011441344987668	2009-12-16	46	Sales Representative
56	Evie	Harrison	evieharrison@examplecom	011441344486508	2023-11-16	46	Sales Representative
57	Scarlett	Gibson	scarlettgibson@examplecom	011441345429268	2030-09-16	47	Sales Representative
58	Ruby	McDonald	rubymcdonald@examplecom	011441345929268	2004-03-16	47	Sales Representative
59	Chloe	Cruz	chloecruz@examplecom	011441345829268	2001-09-16	47	Sales Representative
60	Isabelle	Marshall	isabellemarshall@examplecom	011441345729268	2010-03-16	47	Sales Representative
61	Daisy	Ortiz	daisyortiz@examplecom	011441345629268	2015-12-16	47	Sales Representative
62	Freya	Gomez	freitagomez@examplecom	011441345529268	2003-11-16	47	Sales Representative
63	Phoebe	Murray	phoebemurray@examplecom	011441346129268	2011-11-16	48	Sales Representative
64	Florence	Freeman	florencefreeman@examplecom	011441346229268	2019-03-16	48	Sales Representative
65	Alice	Wells	alicewells@examplecom	011441346329268	2024-09-16	48	Sales Representative
66	Charlotte	Webb	charlottewebb@examplecom	011441346529268	2023-02-16	48	Sales Representative
67	Sienna	Simpson	siennasimpson@examplecom	011441346629268	2024-03-16	48	Sales Representative
68	Matilda	Stevens	matildastevens@examplecom	011441346729268	2021-02-16	48	Sales Representative
69	Evelyn	Tucker	evelyntucker@examplecom	011441343929268	2011-03-16	49	Sales Representative
70	Eva	Porter	evaporter@examplecom	011441343829268	2023-03-16	49	Sales Representative
71	Millie	Hunter	milliehunter@examplecom	011441343729268	2024-09-16	49	Sales Representative
72	Sofia	Hicks	sofiahicks@examplecom	011441343629268	2023-02-16	49	Sales Representative
73	Lucy	Crawford	lucycrawford@examplecom	011441343529268	2024-03-16	49	Sales Representative
74	Elsie	Henry	elsiehenry@examplecom	011441343329268	2021-02-16	49	Sales Representative
75	Imogen	Boyd	imogenboyd@examplecom	011441644429267	2011-05-16	50	Sales Representative
76	Layla	Mason	laylamason@examplecom	011441644429266	2019-03-16	50	Sales Representative
77	Rosie	Morales	rosiemorales@examplecom	011441644429265	2024-03-16	50	Sales Representative
78	Maya	Kennedy	mayakennedy@examplecom	011441644429264	2023-02-16	50	Sales Representative
79	Esme	Warren	esmewarren@examplecom	011441644429263	2024-05-16	50	Sales Representative
80	Elizabeth	Dixon	elizabethdixon@examplecom	011441644429262	2004-09-16	50	Sales Representative

30 rows in set (0.19 sec)

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

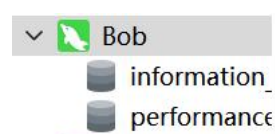
```
mysql> SELECT * FROM sales.salesman
WHERE job_title = 'President';
Empty set
```

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



```
mysql> SELECT * FROM sales.salesman;
1142 - SELECT command denied to user 'Bob'@'localhost' for table 'salesman'
mysql> |
```

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;
```

```
mysql> REVOKE SELECT ON sales.salesman
FROM Alice@localhost;
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)

mysql> |
```

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

观察结果

```
mysql> SELECT * FROM sales.salesman;
1142 - SELECT command denied to user 'Alice'@'localhost' for table 'salesman'
mysql> |
```

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;
```

```
mysql> grant select on sales.salesman to Alice@localhost with grant option;
Query OK, 0 rows affected (0.01 sec)
```

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

```
mysql> grant select on sales.salesman to Bob@localhost;
Query OK, 0 rows affected (0.01 sec)
```



(11)用户 Bob 发布查询语句:

Bob>SELECT \* FROM sales.salesman;

观察系统反馈结果

Objects

pantengkai - Console

Bob - Console

Alice - Console

mysql> SELECT \* FROM sales.salesman;

1142 - SELECT command denied to user 'Bob'@'localhost' for table 'salesman'

mysql> SELECT \* FROM sales.salesman;

	employee_id	first_name	last_name	email	phone	hire_date	manager_id	job_title
	51	Poppy	Jordan	poppyjordan@examplecom	011441344129268	2030-09-16	46	Sales Representative
	52	Sophie	Owens	sophieowens@examplecom	011441344345268	2024-03-16	46	Sales Representative
	53	Sophia	Reynolds	sophiareynolds@examplecom	011441344478968	2020-09-16	46	Sales Representative
	54	Lily	Fisher	lilyfisher@examplecom	011441344498718	2030-03-16	46	Sales Representative
	55	Grace	Ellis	graceellis@examplecom	011441344987668	2009-12-16	46	Sales Representative
	56	Evie	Harrison	evieharrison@examplecom	011441344486508	2023-11-16	46	Sales Representative
	57	Scarlett	Gibson	scarlettgibson@examplecom	011441345429268	2030-09-16	47	Sales Representative
	58	Ruby	Mcdonald	rubymcdonald@examplecom	011441345929268	2004-03-16	47	Sales Representative
	59	Chloe	Cruz	chloecruz@examplecom	011441345829268	2001-09-16	47	Sales Representative
	60	Isabelle	Marshall	isabellemarshall@examplecom	011441345729268	2010-03-16	47	Sales Representative
	61	Daisy	Ortiz	daisyortiz@examplecom	011441345629268	2015-12-16	47	Sales Representative
	62	Freya	Gomez	freyagomez@examplecom	011441345529268	2003-11-16	47	Sales Representative
	63	Phoebe	Murray	phoebemurray@examplecom	011441346129268	2011-11-16	48	Sales Representative
	64	Florence	Freeman	florencefreeman@examplecom	011441346229268	2019-03-16	48	Sales Representative
	65	Alice	Wells	alicewells@examplecom	011441346329268	2024-09-16	48	Sales Representative
	66	Charlotte	Webb	charlottewebb@examplecom	011441346529268	2023-02-16	48	Sales Representative
	67	Sienna	Simpson	siennasimpson@examplecom	011441346629268	2024-03-16	48	Sales Representative
	68	Matilda	Stevens	matildastevens@examplecom	011441346729268	2021-02-16	48	Sales Representative
	69	Evelyn	Tucker	evelyntucker@examplecom	011441343929268	2011-03-16	49	Sales Representative
	70	Eva	Porter	evaporter@examplecom	011441343829268	2023-03-16	49	Sales Representative
	71	Millie	Hunter	milliehunter@examplecom	011441343729268	2024-09-16	49	Sales Representative
	72	Sofia	Hicks	sofiahicks@examplecom	011441343629268	2023-02-16	49	Sales Representative
	73	Lucy	Crawford	lucycrawford@examplecom	011441343529268	2024-03-16	49	Sales Representative
	74	Elsie	Henry	elsiehenry@examplecom	011441343329268	2021-02-16	49	Sales Representative
	75	Imogen	Boyd	imogenboyd@examplecom	011441644429267	2011-05-16	50	Sales Representative
	76	Layla	Mason	laylamason@examplecom	011441644429266	2019-03-16	50	Sales Representative
	77	Rosie	Morales	rosiemorales@examplecom	011441644429265	2024-03-16	50	Sales Representative
	78	Maya	Kennedy	mayakennedy@examplecom	011441644429264	2023-02-16	50	Sales Representative
	79	Esme	Warren	esmewarren@examplecom	011441644429263	2024-05-16	50	Sales Representative
	80	Elizabeth	Dixon	elizabethdixon@examplecom	011441644429262	2004-09-16	50	Sales Representative

30 rows in set (0.12 sec)

30 rows in set (0.12 sec)

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: 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)

```
mysql> REVOKE SELECT ON sales.salesman
FROM Alice@localhost;
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> REVOKE SELECT ON sales.salesman
FROM Bob@localhost;
1142 - SELECT command denied to user 'Alice'@'localhost' for table 'sales
man'
```

Alice 回收权限失败

换一下顺序：

```
mysql> REVOKE SELECT ON sales.salesman
FROM Bob@localhost;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> REVOKE SELECT ON sales.salesman
FROM Alice@localhost;
Query OK, 0 rows affected (0.04 sec)
```

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

```
mysql> CREATE ROLE salesman_role;
Query OK, 0 rows affected (0.02 sec)
```

(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

```
mysql> show grants for salesman_role;
+-----+
| Grants for salesman_role@%          |
+-----+
| GRANT USAGE ON *.* TO `salesman_role`@`%` |
+-----+
1 row in set (0.07 sec)
```

(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;
```

```
mysql> grant select, update on sales.salesman to salesman_role;
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.03 sec)
```

(4)查看 salesman\_role 角色包含的权限，并与(2)比较

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

```
mysql> show grants for salesman_role;
+-----+
| Grants for salesman_role@% |
+-----+
| GRANT USAGE ON *.* TO `salesman_role`@`%` |
| GRANT SELECT, UPDATE ON `sales`.`salesman` TO `salesman_role`@`%` |
+-----+
2 rows in set (0.11 sec)
```

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;
```

```
mysql> revoke update on sales.salesman from salesman_role;
Query OK, 0 rows affected (0.04 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

(6)查看 salesman\_role 角色包含的权限，并与(4)比较

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

```
mysql> show grants for salesman_role;
+-----+
| Grants for salesman_role@% |
+-----+
| GRANT USAGE ON *.* TO `salesman_role`@`%` |
| GRANT SELECT ON `sales`.`salesman` TO `salesman_role`@`%` |
+-----+
2 rows in set (0.09 sec)
```

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 database\_name.table\_name；且用户名格式等同创建时格式

```
mysql> grant salesman_role to Alice@localhost;
Query OK, 0 rows affected (0.03 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

(8)检验角色授权是否生效

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

```
mysql> select * from sales.salesman;
1142 - SELECT command denied to user 'Alice'@'localhost' for table 'salesman'
mysql>
```

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

```
.mysql> set role all;  
{Query OK, 0 rows affected (0.00 sec)}
```

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

employee_id	first_name	last_name	email	phone	hire_date	manager_id	job_title
51	Poppy	Jordan	poppyjordan@examplecom	011441344129268	2030-09-16	46	Sales Representative
52	Sophie	Owens	sophieowens@examplecom	011441344345268	2024-03-16	46	Sales Representative
53	Sophia	Reynolds	sophiareynolds@examplecom	011441344478968	2020-09-16	46	Sales Representative
54	Lily	Fisher	lilyfisher@examplecom	011441344498718	2030-03-16	46	Sales Representative
55	Grace	Ellis	graceellis@examplecom	011441344987668	2009-12-16	46	Sales Representative
56	Evie	Harrison	evieharrison@examplecom	011441344486508	2023-11-16	46	Sales Representative
57	Scarlett	Gibson	scarlettgibson@examplecom	011441345429268	2030-09-16	46	Sales Representative
58	Ruby	Mcdonald	rubymcdonald@examplecom	011441345929268	2004-03-16	47	Sales Representative
59	Chloe	Cruz	chloecruz@examplecom	011441345829268	2001-09-16	47	Sales Representative
60	Isabelle	Marshall	isabellamarshall@examplecom	011441345729268	2010-03-16	47	Sales Representative
61	Daisy	Ortiz	daisyortiz@examplecom	011441345629268	2015-12-16	47	Sales Representative
62	Freya	Gomez	freyagomez@examplecom	011441345529268	2003-11-16	47	Sales Representative
63	Phoebe	Murray	phoebemurray@examplecom	011441346129268	2011-11-16	48	Sales Representative
64	Florence	Freeman	florencefreeman@examplecom	011441346229268	2019-03-16	48	Sales Representative
65	Alice	Wells	alicewells@examplecom	011441346329268	2024-09-16	48	Sales Representative
66	Charlotte	Webb	charlottewebb@examplecom	011441346529268	2023-02-16	48	Sales Representative
67	Sienna	Simpson	siennasimpson@examplecom	011441346629268	2024-03-16	48	Sales Representative
68	Matilda	Stevens	matildastevens@examplecom	011441346729268	2021-02-16	48	Sales Representative
69	Evelyn	Tucker	evelyntucker@examplecom	011441343929268	2011-03-16	49	Sales Representative
70	Eva	Porter	evaporter@examplecom	011441343829268	2023-03-16	49	Sales Representative
71	Millie	Hunter	milliehunter@examplecom	011441343729268	2024-09-16	49	Sales Representative
72	Sofia	Hicks	sofiahicks@examplecom	011441343629268	2023-02-16	49	Sales Representative
73	Lucy	Crawford	lucycrawford@examplecom	011441343529268	2024-03-16	49	Sales Representative
74	Elsie	Henry	elsiehenry@examplecom	011441343329268	2021-02-16	49	Sales Representative
75	Imogen	Boyd	imogenboyd@examplecom	011441644429267	2011-05-16	50	Sales Representative
76	Layla	Mason	laylamason@examplecom	011441644429266	2019-03-16	50	Sales Representative
77	Rosie	Morales	rosiemorales@examplecom	011441644429265	2024-03-16	50	Sales Representative
78	Maya	Kennedy	mayakennedy@examplecom	011441644429264	2023-02-16	50	Sales Representative
79	Esme	Warren	esmewarren@examplecom	011441644429263	2024-05-16	50	Sales Representative
80	Elizabeth	Dixon	elizabethdixon@examplecom	011441644429262	2004-09-16	50	Sales Representative

30 rows in set (0.11 sec)

## 关闭当前会话的生效角色

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

```
mysql> set role none;  
Query OK, 0 rows affected (0.00 sec)
```

## (9)删除角色

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

```
mysql> DROP ROLE salesman_role;  
Query OK, 0 rows affected (0.04 sec)
```

## (10)删除用户

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

```
mysql> DROP USER Alice@localhost;  
Query OK, 0 rows affected (0.04 sec)  
mysql> DROP USER Bob@localhost;  
Query OK, 0 rows affected (0.04 sec)
```

## 3. 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。

2. 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。