目录

[MySQL 创建三权分立账户指南 2](#_Toc22239)

[1. 三权分立概述 2](#_Toc10574)

[1.1 权限分离原则 2](#_Toc13296)

[2. 环境准备 2](#_Toc20011)

[2.1 前提条件 2](#_Toc20805)

[2.2 连接数据库 2](#_Toc4619)

[2.3 检查当前用户和权限 2](#_Toc29483)

[3. 创建系统管理员账户 3](#_Toc8292)

[3.1 创建系统管理员用户 3](#_Toc13851)

[3.2 分配系统管理权限 3](#_Toc12096)

[3.3 限制系统管理员权限 3](#_Toc7587)

[4. 创建审计管理员账户 4](#_Toc27382)

[4.1 创建审计管理员用户 4](#_Toc1972)

[4.2 分配审计管理权限 4](#_Toc30920)

[4.3 配置审计日志 5](#_Toc3926)

[5. 创建安全管理员账户 8](#_Toc3899)

[5.1 创建安全管理员用户 8](#_Toc531)

[5.2 分配安全管理权限 9](#_Toc23139)

[5.3 限制安全管理员权限 9](#_Toc26006)

[6. 权限验证与测试 9](#_Toc21383)

[6.1 测试系统管理员权限 9](#_Toc14153)

[6.2 测试审计管理员权限 10](#_Toc11959)

[6.3 测试安全管理员权限 10](#_Toc7682)

[8. 安全配置建议 11](#_Toc22098)

[7.1 密码策略配置 11](#_Toc17918)

[7.2 连接安全配置 11](#_Toc27540)

[7.3 审计配置优化 11](#_Toc8628)

[8. 常见问题与注意事项 12](#_Toc15513)

[8.1 权限最小化原则 12](#_Toc24141)

[8.2 账户安全管理 12](#_Toc25468)

[8.3 审计日志管理 12](#_Toc4284)

[8.4 备份与恢复 12](#_Toc8065)

[8.5 监控和告警 12](#_Toc2261)

# MySQL 创建三权分立账户指南

本文档详细介绍如何在 MySQL 数据库中创建符合等保三级要求的三权分立账户体系，确保数据库的安全管理。

## 1. 三权分立概述

根据等保三级要求，数据库管理需要实现三权分立，主要包括：

**系统管理员（System Administrator）**：负责数据库系统的安装、配置、维护和日常运维

**审计管理员（Audit Administrator）**：负责审计策略制定、审计日志管理和安全事件分析

**安全管理员（Security Administrator）**：负责用户权限管理、安全策略配置和访问控制

### 1.1 权限分离原则

每个管理员只拥有完成其职责所需的最小权限

关键操作需要多人协作完成

审计功能独立于其他管理功能

确保职责分离，避免权限过度集中

## 2. 环境准备

### 2.1 前提条件

MySQL 5.7 或更高版本

具有 root 权限的 MySQL 管理员账户

已启用 MySQL 审计插件（如 MySQL Enterprise Audit 或 MariaDB Audit Plugin）

### 2.2 连接数据库

# 使用root账户连接MySQL

mysql -u root -p

### 2.3 检查当前用户和权限

-- 查看当前用户

SELECT USER(), CURRENT\_USER();

-- 查看所有用户

SELECT user, host FROM mysql.user;

-- 查看当前用户权限

SHOW GRANTS;

## 3. 创建系统管理员账户

系统管理员主要负责数据库的系统级管理，包括安装、配置、备份恢复等。

### 3.1 创建系统管理员用户

-- 创建系统管理员账户

CREATE USER 'sysadmin'@'localhost' IDENTIFIED BY 'SysAdmin@2024!';

CREATE USER 'sysadmin'@'%' IDENTIFIED BY 'SysAdmin@2024!';

-- 设置密码策略（可选）

ALTER USER 'sysadmin'@'localhost' PASSWORD EXPIRE INTERVAL 90 DAY;

ALTER USER 'sysadmin'@'%' PASSWORD EXPIRE INTERVAL 90 DAY;

### 3.2 分配系统管理权限

-- 授予系统管理相关权限

GRANT SELECT, INSERT, UPDATE, DELETE ON \*.\* TO 'sysadmin'@'localhost';

GRANT SELECT, INSERT, UPDATE, DELETE ON \*.\* TO 'sysadmin'@'%';

-- 授予数据库管理权限

GRANT CREATE, DROP, ALTER, INDEX ON \*.\* TO 'sysadmin'@'localhost';

GRANT CREATE, DROP, ALTER, INDEX ON \*.\* TO 'sysadmin'@'%';

-- 授予备份恢复权限

GRANT LOCK TABLES, RELOAD, PROCESS ON \*.\* TO 'sysadmin'@'localhost';

GRANT LOCK TABLES, RELOAD, PROCESS ON \*.\* TO 'sysadmin'@'%';

-- 授予复制相关权限（如果需要主从复制）

GRANT REPLICATION SLAVE, REPLICATION CLIENT ON \*.\* TO 'sysadmin'@'localhost';

GRANT REPLICATION SLAVE, REPLICATION CLIENT ON \*.\* TO 'sysadmin'@'%';

-- 刷新权限

FLUSH PRIVILEGES;

### 3.3 限制系统管理员权限

-- 系统管理员不应具有以下权限：

-- 1. 用户管理权限（CREATE USER, DROP USER, GRANT OPTION）

-- 2. 审计日志访问权限

-- 3. 安全相关表的修改权限

-- 验证权限分配

SHOW GRANTS FOR 'sysadmin'@'localhost';

## 4. 创建审计管理员账户

审计管理员负责审计策略制定和审计日志管理。

### 4.1 创建审计管理员用户

-- 创建审计管理员账户

CREATE USER 'auditadmin'@'localhost' IDENTIFIED BY 'AuditAdmin@2024!';

CREATE USER 'auditadmin'@'%' IDENTIFIED BY 'AuditAdmin@2024!';

-- 设置密码策略

ALTER USER 'auditadmin'@'localhost' PASSWORD EXPIRE INTERVAL 90 DAY;

ALTER USER 'auditadmin'@'%' PASSWORD EXPIRE INTERVAL 90 DAY;

### 4.2 分配审计管理权限

-- 授予审计日志查看权限

GRANT SELECT ON mysql.general\_log TO 'auditadmin'@'localhost';

GRANT SELECT ON mysql.general\_log TO 'auditadmin'@'%';

GRANT SELECT ON mysql.slow\_log TO 'auditadmin'@'localhost';

GRANT SELECT ON mysql.slow\_log TO 'auditadmin'@'%';

-- 如果使用MySQL Enterprise Audit

-- GRANT AUDIT\_ADMIN ON \*.\* TO 'auditadmin'@'localhost';

-- GRANT AUDIT\_ADMIN ON \*.\* TO 'auditadmin'@'%';

-- 授予性能监控权限

GRANT SELECT ON performance\_schema.\* TO 'auditadmin'@'localhost';

GRANT SELECT ON performance\_schema.\* TO 'auditadmin'@'%';

-- 授予查看进程权限

GRANT PROCESS ON \*.\* TO 'auditadmin'@'localhost';

GRANT PROCESS ON \*.\* TO 'auditadmin'@'%';

-- 刷新权限

FLUSH PRIVILEGES;

### 4.3 配置审计日志

#### 4.3.1 MySQL配置文件设置

-- 启用审计日志（需要在my.cnf中配置）

-- 在MySQL配置文件中添加：

-- [mysqld]

-- general\_log = ON

-- general\_log\_file = /var/log/mysql/mysql.log

-- slow\_query\_log = ON

-- slow\_query\_log\_file = /var/log/mysql/mysql-slow.log

-- long\_query\_time = 2

#### 4.3.2 动态启用日志记录

# 注意：执行前需要先创建日志目录和设置权限

sudo mkdir -p /var/log/mariadb

sudo touch /var/log/mariadb/general.log /var/log/mariadb/slow.log

sudo chown -R mysql:mysql /var/log/mariadb

sudo chmod 640 /var/log/mariadb/\*.log

-- 开启日志记录

SET GLOBAL general\_log = 'ON'; -- 开启一般查询日志

SET GLOBAL slow\_query\_log = 'ON'; -- 开启慢查询日志

SET GLOBAL general\_log\_file = '/var/log/mariadb/general.log';

SET GLOBAL slow\_query\_log\_file = '/var/log/mariadb/slow.log';

SET GLOBAL long\_query\_time = 2; -- 设置慢查询时间为2秒

SET GLOBAL log\_queries\_not\_using\_indexes = 'ON'; -- 记录未使用索引的查询

SET GLOBAL min\_examined\_row\_limit = 1000; -- 慢查询日志的最小检查行数

SET GLOBAL binlog\_format = 'ROW'; -- 二进制日志格式

#### 4.3.3 配置日志轮转策略

上述日志不会自动分割和清理，需要配置自动轮转策略。推荐使用 logrotate 工具。

##### 4.3.3.1 创建配置文件

# 使用 root 权限创建配置文件

sudo nano /etc/logrotate.d/mysql

##### 4.3.3.2 配置文件内容

/var/log/mariadb/\*.log {

daily # 每日轮转

rotate 180 # 保留半年（等保要求）

size 500M # 防止单文件过大

compress # 压缩节省空间

delaycompress # 延迟压缩

dateext # 使用日期命名

dateformat -%Y%m%d-%H # 包含小时：20250827-14

dateyesterday # 使用准确的日期

missingok

notifempty

create 640 mysql mysql

olddir /var/log/mariadb/archive/ # 归档到专门目录

createolddir 755 mysql mysql

postrotate

/usr/bin/mysqladmin flush-logs 2>/dev/null || true

echo "$(date): MariaDB logs rotated" >> /var/log/mariadb/rotation.log

endscript

}

##### 4.3.3.3 配置参数详解

| 参数 | 说明 | 示例 |
| --- | --- | --- |
| daily | 轮转频率 | daily, weekly, monthly |
| rotate 30 | 保留文件数量 | 保留30个备份文件 |
| size 100M | 按大小轮转 | 文件超过100MB时轮转 |
| compress | 压缩旧文件 | 使用gzip压缩 |
| copytruncate | 复制后清空 | 适用于一直打开的文件 |
| create 640 mysql mysql | 新文件权限 | 权限640，属主mysql |

##### 4.3.3.4 测试配置

**语法检查：**

# 检查配置文件语法

sudo logrotate -d /etc/logrotate.d/mysql

**强制执行测试：**

# 强制执行一次轮转（测试用）

sudo logrotate -f /etc/logrotate.d/mysql

# 详细模式测试

sudo logrotate -v -f /etc/logrotate.d/mysql

##### 4.3.3.5 生产环境配置示例

# /etc/logrotate.d/mysql-audit

/var/log/mysql/general.log /var/log/mysql/slow.log /var/log/mysql/error.log {

daily

rotate 90 # 保留3个月

compress

delaycompress

missingok

notifempty

create 640 mysql mysql

sharedscripts

# 轮转前执行

prerotate

# 记录轮转时间

echo "$(date): Starting log rotation" >> /var/log/mysql/rotation.log

endscript

# 轮转后执行

postrotate

# 刷新MySQL日志

if [ -f /var/run/mysqld/mysqld.pid ]; then

/usr/bin/mysqladmin --defaults-file=/etc/mysql/debian.cnf flush-logs

fi

echo "$(date): Log rotation completed" >> /var/log/mysql/rotation.log

endscript

}

##### 4.3.3.6 自动化运行

**查看 logrotate 计划任务：**

# 查看系统的 logrotate 计划任务

cat /etc/cron.daily/logrotate

# 通常内容如下：

#!/bin/sh

/usr/sbin/logrotate /etc/logrotate.conf

**手动触发轮转：**

# 执行所有 logrotate 配置

sudo /usr/sbin/logrotate /etc/logrotate.conf

##### 4.3.3.7 监控和验证

**检查轮转状态：**

# 查看 logrotate 状态文件

cat /var/lib/logrotate/status | grep mysql

# 查看轮转后的文件

ls -la /var/log/mysql/

**预期的文件结构：**

/var/log/mysql/

├── general.log # 当前日志

├── general.log.1 # 昨天的日志（未压缩）

├── general.log.2.gz # 前天的日志（已压缩）

├── general.log.3.gz # 更早的日志

└── slow.log # 当前慢查询日志

配置完成后，MySQL 的审计日志会自动按照策略进行轮转和清理，避免磁盘空间不足的问题，同时满足等保三级的日志保留要求。

#### 4.3.4 查看审计配置

-- 查看日志相关配置

SHOW VARIABLES LIKE '%log%';

SHOW VARIABLES LIKE 'long\_query\_time';

## 5. 创建安全管理员账户

安全管理员负责用户权限管理和安全策略配置。

### 5.1 创建安全管理员用户

-- 创建安全管理员账户

CREATE USER 'secadmin'@'localhost' IDENTIFIED BY 'SecAdmin@2024!';

CREATE USER 'secadmin'@'%' IDENTIFIED BY 'SecAdmin@2024!';

-- 设置密码策略

ALTER USER 'secadmin'@'localhost' PASSWORD EXPIRE INTERVAL 90 DAY;

ALTER USER 'secadmin'@'%' PASSWORD EXPIRE INTERVAL 90 DAY;

### 5.2 分配安全管理权限

-- 授予用户管理权限

GRANT CREATE USER ON \*.\* TO 'secadmin'@'localhost' WITH GRANT OPTION;

GRANT CREATE USER ON \*.\* TO 'secadmin'@'%' WITH GRANT OPTION;

-- 授予权限管理相关权限

GRANT SELECT, INSERT, UPDATE, DELETE ON mysql.user TO 'secadmin'@'localhost';

GRANT SELECT, INSERT, UPDATE, DELETE ON mysql.user TO 'secadmin'@'%';

GRANT SELECT, INSERT, UPDATE, DELETE ON mysql.db TO 'secadmin'@'localhost';

GRANT SELECT, INSERT, UPDATE, DELETE ON mysql.db TO 'secadmin'@'%';

GRANT SELECT, INSERT, UPDATE, DELETE ON mysql.tables\_priv TO 'secadmin'@'localhost';

GRANT SELECT, INSERT, UPDATE, DELETE ON mysql.tables\_priv TO 'secadmin'@'%';

-- 授予角色管理权限（MySQL 8.0+）

-- GRANT ROLE\_ADMIN ON \*.\* TO 'secadmin'@'localhost';

-- GRANT ROLE\_ADMIN ON \*.\* TO 'secadmin'@'%';

-- 刷新权限

FLUSH PRIVILEGES;

### 5.3 限制安全管理员权限

-- 安全管理员不应具有以下权限：

-- 1. 直接的数据操作权限（除权限管理相关表外）

-- 2. 系统管理权限（如RELOAD, SHUTDOWN等）

-- 3. 审计日志的修改权限

-- 验证权限分配

SHOW GRANTS FOR 'secadmin'@'localhost';

## 6. 权限验证与测试

### 6.1 测试系统管理员权限

# 使用系统管理员账户登录

mysql -u sysadmin -p

-- 测试数据库操作权限

CREATE DATABASE test\_sysadmin;

USE test\_sysadmin;

CREATE TABLE test\_table (id INT PRIMARY KEY, name VARCHAR(50));

INSERT INTO test\_table VALUES (1, 'test');

SELECT \* FROM test\_table;

DROP DATABASE test\_sysadmin;

-- 尝试用户管理操作（应该失败）

-- CREATE USER 'test\_user'@'localhost' IDENTIFIED BY 'password'; -- 应该报错

### 6.2 测试审计管理员权限

# 使用审计管理员账户登录

mysql -u auditadmin -p

-- 测试审计日志查看权限

SELECT \* FROM mysql.general\_log LIMIT 10;

SELECT \* FROM mysql.slow\_log LIMIT 10;

-- 查看当前连接

SHOW PROCESSLIST;

-- 尝试数据操作（应该失败）

-- CREATE DATABASE test\_audit; -- 应该报错

### 6.3 测试安全管理员权限

# 使用安全管理员账户登录

mysql -u secadmin -p

-- 测试用户管理权限

CREATE USER 'test\_user'@'localhost' IDENTIFIED BY 'TestUser@2024!';

GRANT SELECT ON mysql.user TO 'test\_user'@'localhost';

SHOW GRANTS FOR 'test\_user'@'localhost';

DROP USER 'test\_user'@'localhost';

-- 查看用户权限信息

SELECT user, host, authentication\_string FROM mysql.user;

-- 尝试数据库操作（应该失败）

-- CREATE DATABASE test\_security; -- 应该报错

## 7. 安全配置建议

### 7.1 密码策略配置

-- 配置密码验证插件（MySQL 5.7+）

INSTALL PLUGIN validate\_password SONAME 'validate\_password.so';

-- 设置密码策略

SET GLOBAL validate\_password\_policy = 'STRONG';

SET GLOBAL validate\_password\_length = 12;

SET GLOBAL validate\_password\_number\_count = 2;

SET GLOBAL validate\_password\_special\_char\_count = 2;

-- 查看密码策略设置

SHOW VARIABLES LIKE 'validate\_password%';

### 7.2 连接安全配置

-- 限制连接数

SET GLOBAL max\_connections = 200;

SET GLOBAL max\_user\_connections = 50;

-- 配置连接超时

SET GLOBAL wait\_timeout = 3600;

SET GLOBAL interactive\_timeout = 3600;

-- 查看连接配置

SHOW VARIABLES LIKE '%timeout%';

SHOW VARIABLES LIKE '%connection%';

### 7.3 审计配置优化

-- 配置详细的审计日志

SET GLOBAL general\_log\_file = '/var/log/mysql/general.log';

SET GLOBAL slow\_query\_log\_file = '/var/log/mysql/slow.log';

SET GLOBAL log\_output = 'FILE';

-- 启用二进制日志（用于恢复和复制）

-- 在my.cnf中配置：

-- log-bin = /var/log/mysql/mysql-bin.log

-- binlog\_format = ROW

-- expire\_logs\_days = 7

## 8. 常见问题与注意事项

### 8.1 权限最小化原则

每个账户只分配完成其职责所需的最小权限

定期审查和清理不必要的权限

避免使用通配符权限（如 \*.\*），尽可能精确到具体数据库和表

### 8.2 账户安全管理

定期更换密码

启用账户锁定策略

监控异常登录行为

及时删除不再使用的账户

### 8.3 审计日志管理

确保审计日志的完整性和不可篡改性

定期备份审计日志

设置日志轮转策略，避免日志文件过大

建立日志分析和告警机制

### 8.4 备份与恢复

# 定期备份用户权限信息

mysqldump -u root -p --single-transaction --routines --triggers mysql > mysql\_grants\_backup.sql

# 备份业务数据库

mysqldump -u sysadmin -p --single-transaction --routines --triggers business\_db > business\_db\_backup.sql

### 8.5 监控和告警

监控登录失败次数

监控权限变更操作

监控异常的SQL执行

设置关键操作的告警机制

**注意事项：**

本指南中的密码仅为示例，实际使用时请设置复杂且唯一的密码

根据实际网络环境调整主机地址限制

定期审查和更新权限配置

确保遵循组织的安全策略和合规要求

在生产环境中实施前，请在测试环境中充分验证所有配置