基于 MySQL 8.0 细粒度授权：单独授予 KILL 权限的优雅解决方案

原创 GreatSQL 2025-03-05 159

基于 MySQL 8.0 细粒度授权：单独授予 KILL 权限的优雅解决方案

一、引言

作为一名数据库从业者，我在日常工作中经常会遇到一个棘手的问题：如何在保证安全的前提下，让业务团队拥有足够的权限去管理数据库执行的 SQL，尤其是终止那些失控的慢查询或异常线程？这个问题看似简单，却牵涉到权限设计、安全合规以及数据库稳定性等多方面的权衡。今天，我们就来聊聊 MySQL 8.0 如何通过权限体系的革新，特别是对 KILL 权限的细化支持，解决了这一痛点，并为 DBA 和业务开发团队带来了更大的灵活性。

二、从痛点说起：KILL SQL 的权限困境

在 MySQL 实际使用中，业务开发人员常希望拥有终止 SQL 的能力。例如，慢查询占用大量资源，或未优化的批量操作导致负载飙升，此时及时 KILL 问题 SQL 是最直接的解决办法。然而，MySQL 的权限设计限制让这一需求难以实现，开发人员往往只能依赖 DBA 操作，既低效又拖延响应。

在 MySQL 的传统版本（比如 5.7 及更早版本）中，KILL 权限的实现存在以下几个问题：

1. **执行用户限制** 默认情况下，用户只能 KILL 自己创建的线程。若业务程序使用统一账号（如 app_user），开发人员想通过此账号终止 SQL，必须知道其密码并登录数据库执行 KILL。然而，企业安全规范通常禁止程序账号单点登录，因其权限过大（至少包括增删改查），导致开发人员无法直接操作。这形成了一个"权限死结"：想 KILL 无权限，有权限却无法登录。
2. **SUPER 权限过大** 为突破上述限制，允许用户 KILL 其他用户的 SQL，需授予 SUPER 权限。这看似可行，但 SUPER 权限是个“大杀器”：持有者不仅能 KILL 任何线程，还可修改全局参数（如 `innodb_buffer_pool_size`）、执行 `STOP GROUP_REPLICATION` 关闭组复制等操作（甚至会导致中断服务）。此权限通常仅限超级管理员使用，对业务开发而言明显过大，且存在严重安全隐患，DBA 自然不愿授予业务团队。

面对这样的困境，我们不禁会问：难道就没有一个两全其美的办法，既能让业务团队有 KILL SQL 的能力，又不至于权限失控吗？

答案是肯定的，MySQL 8.0 的到来，让这一需求成为了现实。

三、MySQL 8.0 的细粒度权限拆分：KILL 权限独立登场

MySQL 8.0 在权限体系上迎来重大改进，最显著的是对 SUPER 权限的拆分。在 5.7 及之前，SUPER 权限如同一个大包袱，功能过于庞杂。而 8.0 引入了动态权限（即运行时可分配的细化权限），将 SUPER

GreatSQL
关注
380 文章 39 粉丝 122K+ 浏览量
获得了 69 次点赞
内容获得 52 次评论
获得了 65 次收藏

TA的专栏
技术分享
收录 47 篇内容

热门文章
MySQL Update执行流程解读 2022-04-01 3127浏览
MySQL Test Run 测试框架介绍 2022-02-20 2375浏览
技术分享 | Prometheus+Grafana监控MySQL浅析 2022-01-17 2164浏览
技术分享 | 浅谈mysql语法解析调试方法 2021-12-13 1883浏览
GDB技巧：使用终端界面模式 2021-12-14 1774浏览

在线实训环境入口
MySQL在线实训环境
查看详情 »

最新文章
工具分享-通过开源工具 tuning-primer快速巡检MySQL5.7 2025-08-01 13浏览

拆解为多个细粒度权限，其具体内容如下：

权限名	解释	影响的部分MySQL5.7命令示例
SYSTEM_VARIABLES_ADMIN	允许在运行时更改全局系统变量，包括使用 SET GLOBAL 和 SET PERSIST，以及更改全局事务特性。	`SET GLOBAL innodb_buffer_pool_size = 268435456;` `SET GLOBAL TRANSACTION ISOLATION LEVEL REPEATABLE READ;`
SESSION_VARIABLES_ADMIN	允许设置需要特殊权限的受限会话系统变量。	`SET SESSION sql_log_bin = 0;`
REPLICATION_SLAVE_ADMIN	允许启动和停止常规复制，使用 CHANGE REPLICATION SOURCE TO、CHANGE MASTER TO 等语句。	`CHANGE MASTER TO MASTER_HOST='replica_host';`
GROUP_REPLICATION_ADMIN	允许启动和停止组复制（Group Replication）。	`START GROUP_REPLICATION;`
BINLOG_ADMIN	允许通过 PURGE BINARY LOGS 和 BINLOG 语句控制二进制日志。	`PURGE BINARY LOGS TO 'mysql-bin.000010';`
SET_USER_ID	允许在视图或存储程序的 DEFINER 属性中指定任何账户作为有效授权 ID。	`CREATE DEFINER='admin'@'[localhost](http://localhost)' VIEW v1 AS ...;`
CONNECTION_ADMIN	控制客户端连接行为，包括终止其他账户线程、绕过连接限制、离线模式连接和读写操作等。	KILL 12345;
ENCRYPTION_KEY_ADMIN	启用 InnoDB 加密密钥轮换。	（无直接命令，通常涉及加密相关配置）
VERSION_TOKEN_ADMIN	允许执行版本令牌（Version Tokens）功能。	（涉及特定插件，如 version_tokens_lock() 函数）
ROLE_ADMIN	允许授予和撤销角色、使用 WITH ADMIN OPTION 和 ROLES_GRAPHML() 函数。	(MySQL5.7 没有 role 功能，不涉及)

SUPER 被拆分为 10 个动态权限。同时，MySQL 8.0 因新增功能和组件，又引入了 29 个动态权限，构成以下 39 个动态权限列表：

权限	级别	解释	是否为 SUPER 拆出
APPLICATION_PASSWORD_ADMIN	全局	启用双密码管理。	
AUDIT_ABORT_EXEMPT	全局	允许审计日志过滤器阻止的查询继续执行。	
AUDIT_ADMIN	全局	启用审计日志配置。	
AUTHENTICATION_POLICY_ADMIN	全局	启用认证策略管理。	
BACKUP_ADMIN	全局	启用备份管理。	
BINLOG_ADMIN	全局	启用二进制日志控制。	是
BINLOG_ENCRYPTION_ADMIN	全局	启用二进制日志加密的激活和停用。	
CLONE_ADMIN	全局	启用克隆管理。	
CONNECTION_ADMIN	全局	启用连接限制/控制。	是
ENCRYPTION_KEY_ADMIN	全局	启用 InnoDB 密钥轮换。	是
FIREWALL_ADMIN	全局	启用防火墙规则管理（任何用户）。	
FIREWALL_EXEMPT	全局	豁免用户免受防火墙限制。	
FIREWALL_USER	全局	启用防火墙规则管理（自身）。	
FLUSH_OPTIMIZER_COSTS	全局	启用优化器成本重新加载。	
FLUSH_STATUS	全局	启用状态指示器刷新。	
FLUSH_TABLES	全局	启用表刷新。	
FLUSH_USER_RESOURCES	全局	启用用户资源刷新。	
GROUP_REPLICATION_ADMIN	全局	启用组复制控制。	是
INNODB_REDO_LOG_ARCHIVE	全局	启用重做日志归档管理。	
INNODB_REDO_LOG_ENABLE	全局	启用或禁用重做日志记录。	
NDB_STORED_USER	全局	启用在 SQL 节点间共享用户或角色（NDB 集群）。	
PASSWORDLESS_USER_ADMIN	全局	启用无密码用户账户管理。	

CTE查询数据量过大导致MySQL 8.0发生CORE问题解析
2025-07-25 13浏览

GreatSQL函数索引失效分析：排序规则匹配机制
2025-07-23 11浏览

误操作后快速恢复数据 binlog 解析为反向 SQL
2025-07-21 16浏览

GreatSQL优化技巧：使用 FUNCTION 代替标量子查询
2025-07-16 10浏览

目录

- 基于 MySQL 8.0 细粒度授权：单独授...
  - 一、引言
  - 二、从痛点说起：KILL SQL 的权限困境

PERSIST_RO_VARIABLES_ADMIN	全局	启用持久化只读系统变量。	
REPLICATION_APPLIER	全局	作为复制通道的 PRIVILEGE_CHECKS_USER 角色。	
REPLICATION_SLAVE_ADMIN	全局	启用常规复制控制。	是
RESOURCE_GROUP_ADMIN	全局	启用资源组管理。	
RESOURCE_GROUP_USER	全局	启用资源组管理（用户级别）。	
ROLE_ADMIN	全局	启用角色授予或撤销，使用 WITH ADMIN OPTION。	是
SESSION_VARIABLES_ADMIN	全局	启用设置受限会话系统变量。	是
SET_USER_ID	全局	启用设置非自身 DEFINER 值。	是
SHOW_ROUTINE	全局	启用访问存储过程定义。	
SKIP_QUERY_REWRITE	全局	不重写此用户执行的查询。	
SYSTEM_USER	全局	将账户指定为系统账户。	
SYSTEM_VARIABLES_ADMIN	全局	启用修改或持久化全局系统变量。	是
TABLE_ENCRYPTION_ADMIN	全局	启用覆盖默认加密设置。	
TELEMETRY_LOG_ADMIN	全局	启用 HeatWave on AWS 的遥测日志配置。	
TP_CONNECTION_ADMIN	全局	启用线程池连接管理。	
VERSION_TOKEN_ADMIN	全局	启用版本令牌函数的使用。	是
XA_RECOVER_ADMIN	全局	启用 XA RECOVER 执行。	

其中，KILL SQL 可以使用 `CONNECTION_ADMIN` 这个动态权限来单独授权。

SUPER 因权限过大的安全隐患，在 8.0 版本开始已经不建议使用。官方文档明确指出，SUPER 已被标记为即将废弃，推荐尽早迁移至动态权限体系。

> SUPER is a powerful and far-reaching privilege and should not be granted lightly. If an account needs to perform only a subset of SUPER operations, it may be possible to achieve the desired privilege set by instead granting one or more dynamic privileges, each of which confers more limited capabilities. See Dynamic Privilege Descriptions.
>
> **Note**
>
> SUPER is deprecated, and you should expect it to be removed in a future version of MySQL. See Migrating Accounts from SUPER to Dynamic Privileges.

具体到 KILL SQL 的场景，还有一些细节要注意，如果开发或业务 DBA 需要 KILL SQL，除了需要刚才讨论的 `CONNECTION_ADMIN` 权限外，还有一个前提，要求能看到对应的线程，这要求授予 `PROCESS` 权限，这样他执行 `show processlist` 时就可以看到所有用户运行的线程了，而非只能看到自己的线程。这个权限 5.7、8.0 通用，允许用户查看其他用户的线程，授权如下:

```
GRANT PROCESS ON *.* TO 'dev_admin';
```

> 注意，PROCESS 权限允许查看他人 SQL 原文，可能导致敏感信息泄露。建议业务部门自行评估安全风险，例如仅授权高级业务 DBA 查看全局 SHOW PROCESSLIST 并执行 KILL SQL。

拥有 PROCESS 权限的用户可按以下步骤终止问题线程：

1. 执行 SHOW PROCESSLIST; 查看运行线程列表，找到目标线程 ID。(对应 `PROCESS` 权限)
2. 执行 KILL <thread_id>; 终止指定线程。(对应 `CONNECTION_ADMIN` 权限) 例如：

```
SHOW PROCESSLIST;
KILL 12345;
```

这样一来，业务团队就拥有了终止任意 SQL 的能力，而 DBA 也不必担心他们会误操作全局参数或关闭服务器。权限控制变得更加精细，安全性和灵活性得到了兼顾。

四、系统线程的安全防护：SYSTEM_USER 权限的加持

然而，细心的读者可能会提出一个问题：如果用户拥有了 PROCESS 权限，他们会不会一不小心 KILL 掉一些关键的系统线程，比如主从复制线程（Replication Thread）或后台维护线程？这确实是一个合理的担忧，因为系统线程一旦被误杀，可能会导致数据库服务中断，甚至引发数据不一致的风险。

MySQL 8.0 的设计者显然也考虑到了这一点。为了保护系统线程，他们引入了 SYSTEM_USER 权限。这个权限的作用是区分普通用户线程和系统线程，并为后者提供额外的保护。具体规则如下：

- 如果某个线程是由具有 SYSTEM_USER 权限的用户创建的（通常是 MySQL 的内置系统账号，比如复制线程的执行用户），那么普通用户（没有 SYSTEM_USER 权限）无法通过 KILL 命令终止它。
- 只有当前会话也具备 SYSTEM_USER 权限的用户，才能 KILL 掉同样具备 SYSTEM_USER 权限的线程。

这意味着，即便业务用户拥有了 PROCESS 权限，他们也无法干扰数据库的后台线程，比如主从复制线程、事件调度线程（Event Scheduler）或管理员的命令行。这种设计极大地提升了系统的稳定性，避免了权限滥用带来的潜在风险。

这就保证了普通用户无法越界操作，系统的核心功能得到了保护。

五、实战演练：如何优雅地使用 KILL 权限

1. 使用 dbops 搭建一主两从的 MySQL8.0.41 实例

> dbops 是什么，见 http://dbops.cn

2. 创建程序账号 app_user，授予业务库 app_db 的增删改查权限

```
create user app_user identified by 'Dbops@2025';
grant select,insert,update,delete on `app\_db`.* to app_user;
```

3. 创建业务库、业务表，并执行长事务

4. 创建开发主管账号 dev_admin，授予查看全局 SQL 和 KILL SQL 权限

```
create user 'dev_admin' identified by 'Dbops@2025';
GRANT PROCESS ON *.* TO 'dev_admin';
GRANT CONNECTION_ADMIN ON *.* TO 'dev_admin';
```

5. 为后台线程账号（如 admin、repl）授予 SYSTEM_USER 权限，标记为受保护线程，普通用户无法 KILL

```
GRANT SYSTEM_USER on *.* to admin@'127.0.0.1';
GRANT SYSTEM_USER on *.* to admin@'localhost';
GRANT SYSTEM_USER on *.* to repl@'1%';
```

随后，重启复制线程，重新连接admin命令行窗口，让权限生效。

6. 测试 KILL SQL

受 SYSTEM_USER 保护的线程无法被终止。

而普通用户（如 app_user）的线程可被终止。

注意事项

在 MySQL 8.0 中，权限不足的错误信息因操作而异。例如，修改全局变量时：

```
mysql> SET GLOBAL TRANSACTION ISOLATION LEVEL REPEATABLE READ;
ERROR 1227 (42000): Access denied; you need (at least one of) the SUPER or SYSTEM_VARIABLES
```

这明确指出所需权限。

但对于 KILL SQL，若缺少 `CONNECTION_ADMIN` 权限，错误信息却不同：

```
mysql> kill 66;
ERROR 1094 (HY000): Unknown thread id: 66
```

此提示具有误导性，无法区分是用户未被授予 `CONNECTION_ADMIN` ，还是目标线程受 `SYSTEM_USER` 保护所致，测试时这两方面都可以注意一下。

六、总结与展望

MySQL 8.0 的细粒度权限体系，特别是动态权限的引入，有效缓解了传统版本“权限过大或过小”的困境。通过单独授予 `CONNECTION_ADMIN` 和 `PROCESS` 权限，业务团队无需 SUPER 权限即可安全管理异常 SQL 线程。而 `SYSTEM_USER` 权限则为系统核心线程提供了可靠保护，避免误操作导致服务中断。

dbops 默认创建 admin@‘127.0.0.1’、admin@‘localhost’、repl@‘1%’ 等账号。

欢迎讨论：是否在 dbops 1.10 新版本中支持 MySQL 8.4.x 尝鲜，默认为此三账号添加 SYSTEM_USER 权限，以进一步提升系统安全性？

参考:

https://dev.mysql.com/doc/refman/8.0/en/grant.html

https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html

mysql mysql创建数据库 sql优化

评论

分享你的看法，一起交流吧～

相关阅读

ACDU周度精选 | 本周数据库圈热点 + 技术干货分享（2025/7/25期）
墨天轮小助手 470次阅读 2025-07-25 15:54:18

ACDU周度精选 | 本周数据库圈热点 + 技术干货分享（2025/7/17期）
墨天轮小助手 436次阅读 2025-07-17 15:31:18

墨天轮「实操看我的」数据库主题征文活动启动
墨天轮编辑部 379次阅读 2025-07-22 16:11:27

MySQL 9.4 创新版正式发行GA
甲骨文云技术 240次阅读 2025-07-24 10:41:34

深度解析MySQL的半连接转换
听见风的声音 205次阅读 2025-07-14 10:23:00

MySQL 9.4.0 正式发布，支持 RHEL 10 和 Oracle Linux 10

严少安　199次阅读　2025-07-23 01:21:32

索引条件下推和分区--一条SQL语句执行计划的分析

听见风的声音　197次阅读　2025-07-23 09:22:58

null和子查询--not in和not exists怎么选择？

听见风的声音　182次阅读　2025-07-21 08:54:19

MySQL数据库SQL优化案例(走错索引)

陈举超　166次阅读　2025-07-17 21:24:40

使用 MySQL Clone 插件为MGR集群添加节点

黄山谷　163次阅读　2025-07-23 22:04:19