MySQL安全性和访问控制

# 管理MySQL用户账户

## MySQL账户管理

CREATE USER 'username'@'host' [auth\_info] #新增用户

DROP USER 'username'@'host' #删除用户

RENAME USER old TO new; #重命名用户名

注意：不指定host,则默认是'username'@'%'，可以在host中使用子网掩码

## 权限分配

GRANT privileges [权限] #分配账户权限

ON databasename.tablename #权限应用级别（库，表，列）

TO 'username'@'host' [auth\_info]

[REQUIRE SSL(x509)] #使用安全套接SSL连接

[WITH GRANT OPTION] #限制资源占用

|  |
| --- |
| GRANT SELECT ON \*.\* TO 'k12\_read'@'%' IDENTIFIED BY "k12\_p@$$w0rd" WITH GRANT OPTION; |

### 账户权限分类

#管理权限

|  |  |  |
| --- | --- | --- |
| 权限名称 | 权限级别 | 权限说明 |
| FILE | 服务器主机上的文件访问 | 文件访问权限 |
| CREATE TEMPORARY TABLES | 服务器管理 | 创建临时表权限 |
| LOCK TABLES | 服务器管理 | 锁表权限 |
| CREATE USER | 服务器管理 | 创建用户权限 |
| PROCESS | 服务器管理 | 查看进程权限 |
| RELOAD | 服务器管理 | 执行flush-hosts, flush-logs, flush-privileges, flush-status, flush-tables, flush-threads, refresh, reload等命令的权限 |
| REPLICATION CLIENT | 服务器管理 | 复制权限 |
| REPLICATION SLAVE | 服务器管理 | 复制权限 |
| SHOW DATABASES | 服务器管理 | 查看数据库权限 |
| SHUTDOWN | 服务器管理 | 关闭数据库权限 |
| SUPER | 服务器管理 | 执行kill线程权限 |

**#对象权限**

|  |  |  |
| --- | --- | --- |
| **权限** | **权限级别** | **权限说明** |
| CREATE | 数据库、表或索引 | 创建数据库、表或索引权限 |
| DROP | 数据库或表 | 删除数据库或表权限 |
| GRANT OPTION | 数据库、表或保存的程序 | 赋予权限选项 |
| REFERENCES | 数据库或表 |  |
| ALTER | 表 | 更改表，比如添加字段、索引等 |
| DELETE | 表 | 删除数据权限 |
| INDEX | 表 | 索引权限 |
| INSERT | 表 | 插入权限 |
| SELECT | 表 | 查询权限 |
| UPDATE | 表 | 更新权限 |
| CREATE VIEW | 视图 | 创建视图权限 |
| SHOW VIEW | 视图 | 查看视图权限 |
| ALTER ROUTINE | 存储过程 | 更改存储过程权限 |
| CREATE ROUTINE | 存储过程 | 创建存储过程权限 |
| EXECUTE | 存储过程 | 执行存储过程权限 |

### 权限说明符

整个服务器：GRANT ALL和REVOKE ALL;

整个数据库：ON database.\*

特定的表 ：ON database.table

特定的列：

特定的存储过程：

注意：usage权限代表无权限，当要不影响用户权限的前提下更改账户密码，使用ssl进行连接，或者连接限制。可以使用usage，语句如下：

grant usage on \*.\* to account identified by 'new\_password';

grant usage on \*.\* to account require ssl;

grant usage on \*.\* to account with max\_connetcions\_per\_hour 10;

### 显示账户权限

**SHOW** **GRANTS** **FOR** 'root'@'%';

**SHOW** **GRANTS**

### 撤销权限

REVOKE privileges [权限]

ON 权限应用级别

FROM 用户

|  |
| --- |
| **REVOKE** **DELETE** **ON** \*.\* **FROM** 'root'@'localhost'; |

# 权限表结构和内容

在mysql数据库中，有mysql\_install\_db脚本初始化权限表，存储权限的表有：user表，db表，table\_priv表，columns\_priv表，proc\_priv表

**MySQL**存取控制包含2个阶段：

* **阶段1**：服务器检查你是否允许连接。
* **阶段2**：假定你能连接，服务器检查你发出的每个请求。看你是否有足够的权限实施它。例如，如果你从数据库中一个表精选(select)行或从数据库抛弃一个表，服务器确定你对表有**select**权限或对数据库有**drop**权限。

服务器在存取控制的两个阶段使用在mysql的数据库中的user、db和host表，在这些授权表中字段如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **表名称** | user | db | host |
| **范围字段** | Host | Host | Host |
|  | User | Db | Db |
|  | Password | User |  |
| **权限字段** | Select\_priv | Select\_priv | Select\_priv |
|  | Insert\_priv | Insert\_priv | Insert\_priv |
|  | Update\_priv | Update\_priv | Update\_priv |
|  | Delete\_priv | Delete\_priv | Delete\_priv |
|  | Index\_priv | Index\_priv | Index\_priv |
|  | Alter\_priv | Alter\_priv | Alter\_priv |
|  | Create\_priv | Create\_priv | Create\_priv |
|  | Drop\_priv | Drop\_priv | Drop\_priv |
|  | Grant\_priv | Grant\_priv | Grant\_priv |
|  | Reload\_priv |  |  |
|  | Shutdown\_priv |  |  |
|  | Process\_priv |  |  |
|  | File\_priv |  |  |

对存取控制的第二阶段(请求证实)，如果请求涉及表，服务器可以另外参考tables\_priv和columns\_priv表。这些表的字段如下：

|  |  |  |
| --- | --- | --- |
| **表名称** | tables\_priv | columns\_priv |
| **范围字段** | Host | Host |
|  | Db | Db |
|  | User | User |
|  | Table\_name | Table\_name |
|  |  | Column\_name |
| **权限字段** | Table\_priv | Column\_priv |
|  | Column\_priv |  |
| **其他字段** | Timestamp | Timestamp |
|  | Grantor |  |

每个授权表包含范围字段和权限字段。

user表主要分为：用户列、权限列、安全列、资源控制列

# 服务器如何控制客户端访问

## 访问范围列的内容

Host：ipv4,ipv6,域名（通配符%，\_）

User：空白被当做匿名用户

Db：数据库名

Table\_name,Column\_name和Routine\_name：表名，列名，存储列名

Routine\_type：

## 语句访问验证

验证顺序：user or db or tables\_priv or columns\_priv or procs\_priv，其中这些表的内容在服务器启动的时候，就加载在内存中。

**注意**：当使用create user ,drop user ,rename user ,grant ,revoke,set passwor语句建立和修改账户时候，不需要使用FLUSH PRIVILEGES.

## 访问范围列匹配顺序

服务器读取user表的时候，它会根据host和user的值进行排序。以host为主（优先排序），host值相同时候，会根据user来进行排序。原则：相对于模式，服务器更喜欢字面量，相对于特定程度较低模式，服务器喜欢使用特定程度较高的模式。运行优先级顺序如下：

172.16.129.201

172.16.129.\*

172.16.\*

注意：user表中包含匿名用户时候，该host对应的的用户会优先匹配。