MySQL权限管理

一，权限逻辑
MySQL数据库采用的是白名单的权限策略。明确的制定了那些用户能够对MySQL做什么，没有明确的表示某些用户不能做什么，所以MySQL在检查用户连接时主要分为两阶段

二，用户能否连接
MySQL数据库验证权限有三个地方；用户名，用户密码和来源主机，这三项信息的值保存在mysql库中的user表内，分别对应表中的user，password和host三列
其中来源主机是MySQL重要的部分，即使同一个用户名但是登入的主机不同连接，也视为了两个不同的用户

三，能否执行操作
练到数据库之后，可不可以进行操作，比如删库，建表，查询或修改数据等，都会涉及到 mysql.user 、 mysql.db mysql tables_priv \mysql.columns_priv mysql.proc_priv五个字典表，这五个字典表对数据库、表，列、等对象做了详细的控制

其中mysqsl.user 是全局管理，mysql.db是指数据库级别的，mysql.tables_priv是表级别的。
mysql,columns_priv是列级别的，mysql.proc_priv是程序级别的

四，创建用户与权限授予及回收
create user kk@'192.168.78.%' odemtofoed bu '123456';
创建了一个kk用户，来源主机为192.168.78的网段下的主机都是kk用户。

create user kk@'192.168.78.131' identified by '123456'
创建了一个Kk用户，其来源主机是192.168.78.131的都视为kk用户

使用 help grant; 来查看权限授予的语法

grant all on zsdk.* to kk@'192.168.78.%' ;
将zsdk库下的所有对象授权给Kk@192.168.78网段下登入MySQL的用户、

grant select on kk@'192.168.79.&' ;
给kk@192.168.79.&的用户设置没了没有任何权限

grant alll on kk@'192.168.78.&' ;
给kk@192.168.79.&的用户设置全局权限

grant select,insert,update,delete on zsdk.* kk@'192.168.79.&' ;
在zsdk上授权选择、插入、更新、删除。* kk@'192.168.79。& '。

grant select on zsdk.* to kk@'192.168.79.&' ;
给kk@192.168.79.&的用户设置没了只读权限

当这个流程走完之后需要使用flush privi;eges ; 刷新下数据库

show grants ;
查看权限，还可以单独的使用 show grants for kk@'192.168.78.&' ； 单独的查看权限

回收权限
revoke select on zsdk.* from kk@'192.168.78.&' ;

五，限制操作
MAX_QUERIES_PER__HOUR 允许用户每小时执行查询语句的数据量
MAX_UPDATES_PER_HOUR 允许用户每小时执行的更新语句的数量

MAX_CONNECTIONS_PER_HOUR 允许用户每小时的链接的次数
MAX_USER_CONNECTIONS 允许用户同时连接服务器的数量

六，权限级别
全局：mysql.user
指的是能够拥有该MySQL服务器所有的数据库的素哟欧对象，的所有权限，简称全局

库：mysql.db

表：mysql.tables_priv

列：msql.columns_priv

Host 来源主机
user 用户名
db 某个用户可以对他进行操作
column 对某个用户进行列的操作

可以使用desc mysql.user 进行查询权限，但不仅仅只局限于mysql.user，明白我的意思吧~~~

七，用户权限设定与建议
一般库为单位的差固件账户，在达到了安全设定的前提下，可将权限级别分为三级：
1.{user}*oper: 定义为操纵用户，拥有指定库下的所有对象的DML权限，主要用于前端应用程序和连接数据库读写。*
*2.{user}*read:定义为只读用户，拥有指定库下所有的对象的读取权限
3.{user}_mgr:定义为管理账户，拥有库下对象的DDL及DML权限，用于我项目负责人实时操作对象及数据。

需要注意的事情是，在5.7.26之前的版本中，MySQL中的有一个文件 .mysql_history，会储存相关的私密信息，比如你创建用户时所搞到的~