42.6 权　限

mysql 存取控制包含 2 个阶段：

阶段 1：服务器检查是否允许你的 IP 地址连接 mysql 服务器。检查是否允许你从哪连接。

阶段 2：如果可以连接上。 检查你对具体的库，表，字段有没有权限。

这两个阶段使用 mysql 数据库的 user，db，host 表。如果语法涉及表，服务器可以另外参考：table_priv 表和 columns_priv 表

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		

MySQL 权限经验原则：

权限控制主要是出于安全因素，因此需要遵循一下几个经验原则：

只授予能满足需要的最小权限，防止用户干坏事。比如用户只是需要查询，那就只给 select 权限就可以了，不要给用户赋予 update、insert 或者 delete 权限。

2、创建用户的时候限制用户的登录主机，一般是限制成指定 IP 或者内网 IP 段。@’ 192.168.1.28’ hostname

3、初始化数据库的时候删除没有密码的用户。安装完数据库的时候会自动创建一些用户，这些用户默认没有密码。

4、为每个用户设置满足密码复杂度的密码。CRM 疑是地上霜 CRM14floor&& 双

5、定期清理不需要的用户。回收权限或者删除用户。帐号审计

查看：

```
mysql> use mysql;
mysql> select database();
+------------+
| database() |
+------------+
| mysql      |
+------------+
1 row in set (0.00 sec)
mysql> select * from user\G
*************************** 1. row ***************************
          Host: localhost
          User: root
          Password: *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
          Select_priv: Y
          Insert_priv: Y
          Update_priv: Y
          Delete_priv: Y
          Create_priv: Y
            Drop_priv: Y
          Reload_priv: Y
        Shutdown_priv: Y
         Process_priv: Y
            File_priv: Y
           Grant_priv: Y
      References_priv: Y
           Index_priv: Y
           Alter_priv: Y
         Show_db_priv: Y
```

总结：

存储权限的表有：

1、user 表

2、db 表

3、host 表

4、table_priv 表

5、columns_priv 表

每个授权表包含范围字段和权限字段。

MySQL 权限实战：

1、查看用户权限

mysql> select * from mysql.user\G 查看所有

mysql> select * from mysql.user where user='root' and host='localhost'\G 精确查看

查看当前用户权限：

```
mysql> show grants;
+---------------------------------------------------------------------+
| Grants for root@localhost                                           |
+---------------------------------------------------------------------+
| GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION |
+---------------------------------------------------------------------+
```

查看某个用户的权限：

```
mysql> show grants for 'HA'@'localhost';
+-----------------------------------------------------------------------------------------
-------------------------------------------------+
| Grants for HA@localhost
                                                 |
+-----------------------------------------------------------------------------------------
-------------------------------------------------+
| GRANT ALL PRIVILEGES ON *.* TO 'HA'@'localhost' IDENTIFIED BY PASSWORD '*6BB4837E
B74329105EE4568DDA7DC67ED2CA2AD9' WITH GRANT OPTION |
+-----------------------------------------------------------------------------------------
-------------------------------------------------+
```

mysql 创建用户并授权---grant 的用法

格式：grant 权限 on 数据库名.表名 to 用户@登录主机 identified by "用户密码";
grant[英][grɑ:nt] 承认; 同意; 准许; 授予;

例1：允许tree用户从localhost登录
mysql> grant all on book.* to tree@localhost identified by "123456";

grant 命令说明：

ALL PRIVILEGES 是表示所有权限，你也可以使用 select、update 等权限

ON 用来指定权限针对哪些库和表。

***.* 中前面的*号用来指定数据库名，后面的*号用来指定表名。**

TO 表示将权限赋予某个用户。

tree@'localhost'表示 HA 用户，@后面接限制的主机，可以是 IP、IP 段、域名以及%，%表示任何地方。注意：这里%有的版本不包括本地，以前碰到过给某个用户设置了%允许任何地方登录，但是在本地登录不了，这个和版

本有关系，遇到这个问题再加一个 localhost 的用户就可以了。

IDENTIFIED BY 指定用户的登录密码。

WITH GRANT OPTION 这个选项表示该用户可以将自己拥有的权限授权给别人。注意：经常有人在创建操作用户的时候不指定 WITH GRANT OPTION 选项导致后来该用户不能使用 GRANT 命令创建用户或者给其它用户授权。

备注：可以使用 GRANT 重复给用户添加权限，权限叠加，比如你先给用户添加一个 select 权限，然后又给用户添加一个 insert 权限，那么该用户就同时拥有了 select 和 insert 权限。

允许tree2用户从任意远端主机连接mysql服务器：

```
mysql> grant all privileges on *.* to tree2@'%' identified by '123456' with grant option;
```

with grant option 意思是 tree2 用户可以把自己的权限下放给新创建的用户。 另外，加不加 privileges 都可以。

测试：

```
[root@xuegod64 ~]#  mysql -u tree2 -h 192.168.1.63 -p123456
mysql>    #登录正常
```

但是：

```
[root@xuegod63 ~]# mysql -u tree2 -h 192.168.1.63 -p123456  #不能登录
```

解决方法：

```
mysql>  grant all privileges on *.* to 'tree2'@'192.168.1.63' identified by '123456' with grant option;
```

```
[root@xuegod63 ~]# mysql -u tree2  -p123456   #不能登录
```

解决方法：

```
mysql>  grant all privileges on *.* to 'tree2'@'localhost' identified by '123456' with grant option;
```

总结： % 指的是任意远程主机，不包括本地地址和localhost。 另外grant是立即生效。不需要执行：mysql> flush privileges;

只有手动修改了mysql相关字段，才需要执行mysql> flush privileges;

只授权部分权限：

```
mysql> grant select,insert,update,delete,create,drop on aa.* to 'custom'@'localhost' identified by '123456';
```

方法二： 直接修改表中权限文件：

```
mysql> use mysql ;
mysql> insert into user (Host,User,Password) values('localhost','grace','123456');
```

```
mysql> select Host,User,Password from user where User="grace";
+-----------+-------+----------+
| Host      | User  | Password |
+-----------+-------+----------+
| localhost | grace | 123456   |
+-----------+-------+----------+
```

可以看到密码是明文存放的，现在以加密方式存储：

```
mysql>  insert into user (Host,User,Password) values('localhost','grace1',password("123456"));
Query OK, 1 row affected, 3 warnings (0.00 sec)

mysql> select Host,User,Password from user where User="grace1";
+-----------+--------+-------------------------------------------+
| Host      | User   | Password                                  |
+-----------+--------+-------------------------------------------+
| localhost | grace1 | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
+-----------+--------+-------------------------------------------+
1 row in set (0.01 sec)

mysql> flush privileges;  #刷新权限表，使配置文件生效
```

或重启mysql 数据库

使用这个命令使权限生效，尤其是你对那些权限表 user、db、host 等做了 update 或者 delete 更新的时候。以前遇到过使用 grant 后权限没有更新的情况，只要对权限做了更改就使用 FLUSH PRIVILEGES 命令来刷新权限。

```
mysql> flush privileges;


[root@xuegod63 ~]# service mysqld restart
```

测试：

```
[root@xuegod63 ~]# mysql -u grace  -p123456  #登录不成功
ERROR 1045 (28000): Access denied for user 'grace'@'localhost' (using password: YES)
[root@xuegod63 ~]# mysql -u grace1  -p123456  #登录成功
```

回收权限

```
mysql> revoke delete on *.* from 'tree'@'localhost';
```

42.7 修改 mysql 用户密码

42.7.1 grant 修改立即生效

```
mysql>  grant all on book.* to mk1@localhost identified by "123456";
mysql> grant select,insert,update,delete,create,drop on aa.* to 'custom'@'localhost' identified by '123456';
```

测试：

```
[root@xuegod63 ~]# mysql -u mk1 -p123456
[root@xuegod63 ~]# mysql -u custom -p123456
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| book               |
| test               |
```

设置帐户密码：

```
[root@xuegod63 ~]# mysqladmin -u root -p
```

42.7.2 set password

```
mysql> set password for mka@"localhost" = password('456789');
mysql> FLUSH PRIVILEGES;
```

42.7.3 重置 mysql root 密码

方法一：

```
[root@xuegod63 mysql]# /etc/init.d/mysqld stop
[root@xuegod63 mysql]# mysqld_safe --skip-grant-tables --skip-networking
```

在另一个终端打开：

```
mysql> update mysql.user set password=password('1234567') where host=' localhost' and user='root';
[root@xuegod63 aa]# /etc/init.d/mysqld restart
```

本节课的考题：

1、 删除了所有用户怎么办

mysql> delete from mysql.user; 删除所有用户

mysql> flush privileges;

没有用户可以登录

解决办法：

service mysqld stop

mysqld_safe --skip-grant-tables --skip-networking

另外终端

Mysql

mysql> insert into mysql.user (host, user, password) values ('localhost', 'root', password('123456'));

mysql> flush privileges;

mysql> grant all privileges on *.* to 'root'@'localhost';

/etc/init.d/mysqld restart

或者

Mysql 登陆后

mysql> flush privileges; 先刷新表，一定要先刷新

不能出现这个错误：

```
mysql> grant all privileges on *.* to system@localhost identified by '123456';
ERROR 1290 (HY000): The MySQL server is running with the --skip-grant-tables option
 so it cannot execute this statement
```

mysql> grant all privileges on *.* to system@localhost identified by '123456' with grant option;

mysql> flush privileges;

/etc/init.d/mysqld restart
mysql -usystem -p123456

mysql 日常维护工具：
开始之前，把 xuegod63 和 xuegod64 都恢复：
[root@xuegod63 ~]# yum install mysql-server -y
[root@xuegod63 ~]# service mysqld restart

上传 book.sql 到 xuegod63,然后导入数据库
[root@xuegod63 ~]# mysql

mysql> create database book;
[root@xuegod63 ~]# mysql -uroot -p book < book.sql
Enter password:

[root@xuegod63 ~]# mysqladmin -u root password "123456"

mysql 修复工具 mysqlcheck

什么时候用这个工具？
举例：
数据库服务器（mysql）因机房掉电，异常关机。等服务器重新起来，启动mysql服务后，发现部分表有所损坏。报can't open file: "xxxx.MYI"(errno: 145)

mysqlcheck使用

mysqlcheck客户端工具可以检查和修复MyISAM表，还可以优化和分析表。
实际上，它集成了mysql工具中check、repair、analyze、optimize的功能。
analyze[英]['ænəlaɪz] 分析 optimize[英][ˈɒptɪmaɪz] 优化

/usr/local/mysql/bin/mysqlcheck #源码编译安装位置
rpm -qf `which mysqlcheck` yum 安装查看

参数选项
option中有以下常用选项：
-c, --check (检查表)；
-r, --repair（修复表)；
-a, --analyze (分析表)；
-o, --optimize(优化表)； //其中，默认选项是-c(检查表)
-u， 使用 mysql 中哪个用户进行操作

mysqlcheck 语法：
使用以下3种方式来调用mysqlcheck：
#mysqlcheck[options] db_name [tables]
mysqlcheck[options] ---database DB1 [DB2 DB3...]
#mysqlcheck[options] --all--database

如果没有指定任何表或使用---database或--all--database选项，则检查整个数据库。

例:
[root@xuegod63 ~]# rpm -qf `which mysqlcheck `
mysql-5.1.71-1.el6.x86_64
检查表（check）;
[root@xuegod63 ~]# mysqlcheck -u root -p123456 -c book books
book.books   OK

修复表（repair）；

 [root@xuegod63 ~]# mysqlcheck -u root -p123456 -c book books
book.books   OK

修复指定的数据库
[root@xuegod63 ~]# mysqlcheck -uroot -p -r --database book
Enter password:
book.books   OK
book.category   OK
参数：
-B, --databases   Check several databases

检查修复所有的数据库
[root@xuegod63 ~]# mysqlcheck -u root -A -r -p
Enter password:
book.books   OK
book.category   OK
mysql.columns_priv   OK
参数-A 等于 --all-databases

实战1：每天定时自动优化MySQL数据库

[root@xuegod63 ~]# crontab -e   //把它加入 cron job 每天在 01:00 自动执行
#0 1 * * * mysqlcheck -A -o -r -u你的用户名 -p你的密码 > /dev/null 2>&1

0 3 * * * mysqlcheck -uroot -p123456 -r -o -A > /dev/null 2>&1

mysql备份与恢复：

按照备份时对数据库的影响范围,备份的方法

Hot backup(热备) Cold Backup（冷备） Warm Backup（温备）

Hot backup：指在数据库运行中直接备份，对正在运行的数据库没有任何影响。（Online Backup）官方手册为在线备份

Cold Backup：指在数据库停止的情况下进行备份(OfflineBackup) 官方手册称为离线备份
Warm Backup：备份同样在数据库运行时进行，但是会对当前数据库的操作有所影响，例如加一个全局读锁以保证备份数据的一致性

按照备份后文件内容：

逻辑备份-->

指备份后的文件内容是可读的，通常为文本文件，内容一般是SQL语句，或者是表内的实际数据，如mysqldump和SELECT * INTO OUTFILE的方法，一般适用于数据库的升级和迁移，恢复时间较长

裸文件备份-->
拷贝数据库的物理文件，数据库既可以处于运行状态（mysqlhotcopy 、ibbackup、xtrabackup这类工具），也可以处于停止状态，恢复时间较短。

按照备份数据库的内容来分，又可以分为：

完全备份：对数据库完整的备份

增量备份：在上一次完全备份基础上，对更新的数据进行备份（xtrbackup）

日志备份：二进制日志备份，主从复制

逻辑备份工具：mysqldump
使用的时候 MySQL 当要导入或者导出数据量大的库的时候,用 PHPMYADMIN 甚至 MySQL Administrator 这些工具都会力不从心,这时只能使用 MySQL 所提供的命令行工具 mysqldump 进行备份恢复。数据量大的时候不推荐使用，可支持 MyISAM,InnoDB

MySQL 数据的导出和导入工具:mysqldump。

导出数据：
语法： mysqldump [OPTIONS] database [tables] >导出的文件名.sql

例：1.导出整个 book 数据库
mysqldump -u 用户名 -p 数据库名 > 导出的文件名.sql # 注意是-p 空格后是数据库名，不是密码。
[root@xuegod63 ~]# mysqldump -u root -p book > /opt/book.sql
或：
[root@xuegod63 mysql]# mysqldump -u root -p123456 book > /opt/book.sql
查看内容：
[root@xuegod63 mysql]# vim /opt/book.sql

例：2.导入数据库
法一：
[root@xuegod63 ~]# mysql -u root -p book < /opt/book.sql
Enter password: 输入密码即可。
如果导入时，没有对应的数据库，需要你手动创建一下：
mysql> create database book;

方法二：使用source 命令导入数据
如mysql -u root -p
mysql>use 数据库
然后使用source命令，后面参数为脚本文件（如这里用到的.sql）
mysql> drop table books; #删除数据，再导入数据

mysql> source /opt/book.sql ;

。。。

Query OK, 44 rows affected (0.00 sec) #查询行，44 行的影响（0 秒）

Records: 44 Duplicates: 0 Warnings: 0 #记录：44 份：0 警告：0

。。。

Duplicates ˈdju:plɪkeɪt 重复的

mysql> create database book;
mysql -usystem -p123456 book<booktable.sql 先导入表结构
mysql -usystem -p123456 book<bookdata.sql 再导入数据

c:导入表
mysql> drop table books;
mysql> source /root/books.sql; ##导入表时，不需要重新，创建表。
mysql> select * from books;
D：导入表结构和数据
mysql> create database book;
mysql -usystem -p123456 book<booktable.sql 先导入表结构
mysql -usystem -p123456 book<bookdata.sql 再导入数据

例：导出单张表：

[root@xuegod63 mysql]# mysqldump -u root -p123456 book books > books.sql
#导入book库中的books表。
导入：
mysql> drop table books;
mysql> source /root/books.sql; ##导入表时，不需要重新，创建表。
mysql> select * from books;

例：导出所有数据库：
[root@xuegod63 ~]# mysqldump -u root -p123456 -A > all1.sql
[root@xuegod63 ~]# mysqldump -u root -p123456 --all-databases > all2.sql
或：
参数：-A, --all-databases Dump all the databases.
[root@xuegod63 /]# vim all.sql #查看sql语句，导入数据时，会自动创建对应的数据库

```
CREATE DATABASE /*!32312 IF NOT EXISTS*/ `book` /*!40100 DEFAULT CHARACTER SET lat
in1 */;

USE `book`;
```

导入：

[root@xuegod63 /]# mysql -u root -p123456 < all.sql

D:导出库的表结构

mysqldump -usystem -p123456 -d book>booktable.sql #只导出 book 库的表结构

E：只导出数据

mysqldump -usystem -p123456 -t book>bookdata.sql #只导出 book 库中的数据

F:导出数据库，并自动生成库的创建语句

mysqldump -uroot -p123456 -B book2 >book2.sql

mysql -uroot -p123456 < book2.sql 导入不用指定数据名

学员问题：

GTID 方式会讲吗？

SQLyog 很不错，老师会讲这个工具吗？