**常用操作**

**（mysql连接、增删改查、权限）**

# 如何使用帮助

## 查看命令使用方法

help

查看日志删除使用方法

help purge

## 查看慢日志的参数

show variables like ‘%slow%’;

# 基础操作

## mysql常用操作

## MySQL基础操作

show databases; <--- 查询默认的数据库信息，相当于linux的查看命令ll

create database oldboy; <---创建新的数据库

drop database oldboy; <---删除存在的数据库

use mysql; <--- 表示选择使用一个数据库，相当于cd进入一个数据库

show tables； <---查看数据库中表信息

select database(); <--- 表示查看当前所在数据库，类似于pwd命令的功能

select user(); <--- 查看当前登录数据库的用户，类似于whoami命令

并且mysql还可以限制指定用户可以从哪里进行连接登录数据库

select \* from user\G; <---查看user表中所有信息，并且纵行显示

select user,host from user; <---查看user表中指定信息，并且横行显示，相当于linux的awk命令

select user,host from mysql.user; ---查看可以登录mysql数据库的目录，以及都可以从哪里进行管理mysql数据库

grant all on \*.\* to user@'host' identified by 'oldboy123'; ---创建用户

grant all on \*.\* to Old\_Boy@'localhost' identified by 'oldboy123'; ---创建用户（大写用户）

drop user 'user'@'host';

flush privileges； --- 刷新权限

### 添加删除数据库系列

#创建数据库

create database wordpress;

#删除数据库

drop database wordpress;

### 删除表

#逻辑删除，对文件内容读一行，删除一行，所以删除速度慢，适用于小表的删除；如果误删，可以恢复。

DELETE FROM student； #删除较小的表，不标准写法

DELETE FROM student WHERE sid=3 #标准的写法

#从物理层面删除，将表删除，删除速度快，适用于大表删除；如果误删，只能找专业的数据公司恢复，且可能恢复得不完整

TRUNCATE TABLE student;

## 详细：查看数据库表

mysql> use mysql;

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

mysql> show tables;

#信息均来自于information\_schema（元数据的查询视图，中间者，真正的数据存放在系统表中，比如元数据表（也叫基表））中

#==mysql所有的show显示的信息，均来自于information\_schema

+---------------------------+

| Tables\_in\_mysql |

+---------------------------+

| columns\_priv |

| db |

| event |

| func |

| general\_log |

| help\_category |

……

mysql>

## 用户权限管理举例

### 查看用户权限

#查看wordpress用户的权限为select权限

mysql> show grants for wordpress@'10.0.0.%';

| Grants for wordpress@10.0.0.% | GRANT SELECT ON \*.\* TO 'wordpress'@'10.0.0.%' IDENTIFIED BY PASSWORD '\*23AE809DDACAF96AF0FD78ED04B6A265E05AA257'

#查看root@‘localhost’用户的权限为‘ALL PRIVILEGES’权限，并且拥有授权权限“ALL PRIVILEGES”权限

mysql> show grants for root@'localhost'；

| GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'localhost' IDENTIFIED BY PASSWORD '\*FE28814B4A8B3309DAC6ED7D3237ADED6DA1E515' WITH GRANT OPTION |

| GRANT PROXY ON ''@'' TO 'root'@'localhost' WITH GRANT OPTION

#查看root@'10.0.0.%'用户的权限为“ALL PRIVILEGES”权限

mysql> show grants for root@'10.0.0.%';

| Grants for root@10.0.0.%

| GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'10.0.0.%' IDENTIFIED BY PASSWORD '\*FE28814B4A8B3309DAC6ED7D3237ADED6DA1E515' |

### 删除用户权限

#模拟赋予oldboy用户，在oldboy库上的权限为select,update,delete,insert

mysql> grant select,update,delete,insert on oldboy.\* to oldboy@'10.0.0.%' identified by '123';

Query OK, 0 rows affected (0.00 sec)

#查看用户oldboy的权限，确认是否授权完成

mysql> show grants for oldboy@'10.0.0.%';

| Grants for oldboy@10.0.0.%

| GRANT USAGE ON \*.\* TO 'oldboy'@'10.0.0.%' IDENTIFIED BY PASSWORD '\*23AE809DDACAF96AF0FD78ED04B6A265E05AA257' |

| GRANT SELECT, INSERT, UPDATE, DELETE ON `oldboy`.\* TO 'oldboy'@'10.0.0.%'

#取消用户oldboy在库oldboy上的所有权限，

mysql> revoke select,insert,update,delete on oldboy.\* from oldboy@'10.0.0.%';

Query OK, 0 rows affected (0.00 sec)

#查看用户oldboy的权限，确认是否取消完成

mysql> show grants for oldboy@'10.0.0.%';

| GRANT USAGE ON \*.\* TO 'oldboy'@'10.0.0.%' IDENTIFIED BY PASSWORD '\*23AE809DDACAF96AF0FD78ED04B6A265E05AA257' |

## 用户基本管理

### 用户的作用

1）linux用户作用

①登录

②管理系统文件

2）MySQL用户

①登录mysql数据库

②管理数据库对象

### 用户的创建

#### 格式说明

EXP:‘username’@’主机域’

----主机域：可以理解为是MySQL允许登录的白名单

----主机域格式：

→“%”==百分号==表示指定任何一个字符

-------’10.0.0.51’

-------’10.0.0.5%’

#表示10.0.0.50~10.0.0.59均可以连接数据库

-------’10.0.0.%’

#表示10.0.0.%网段地址，默认是24位，如果是23位就使用’10.0.0.0/255.255.255.0’格式

-------’%’#所有人均可以访问，除非防火墙有策略

-------’10.0.0.0/255.255.255.0’#可以指定特殊子码掩码

-------’db01’ #支持的主机名

#### 创建用户

#创建用户

mysql> create user wordpress;

#创建用户并设定改密码

mysql> create user wordpress@’10.0.0.%’ identified by ‘oldboy123’；

### 用户管理实战

#### 安装完系统，设定初始密码（'root'@'localhost'）

第一次设定密码的格式：

[root@db02 ~]# mysqladmin -uroot -p password '123'

第二次设定密码的格式：

[root@db02 ~]# mysqladmin -uroot -p password 'oldboy123'

Enter password:

#### 安装完MySQL，使用密码登录

mysql -uroot -poldboy123

#### 清理无用用户

drop user ''@'db02';

drop user ''@'localhost';

drop user 'root'@'db02';

drop user 'root'@'::1';

#### 查看用户信息

1）mysql> select \* from mysql.user; #查看user表中所有信息

2）select user,host from mysql.user; #查看mysql库下的user表中user和host==绝对路径

3）use mysql；

select user,host from user; #相对路径，意义同上

#### 修改用户密码

方法一：set password（只能修改当前你用户）；

方法二：grant（综合命令，授权+改密码；推荐使用）

方法三：忘了密码才会使用以下命令修改密码

MYSQL5.7以下版本的数据库密码使用的是

mysql这个数据库里的user表的password这个字段，

修改密码只需：

1. update mysql.user set password=PASSWORD('OLDBOY123') where user='root';

2.flush privileges; #退出数据库操作行就生效，不用重启mysql

可是到了5.7版本，user表里就没有了password这个字段了，

要想修改密码则需要用authentication\_string这个字段：

1.update mysql.user set authentication\_string=PASSWORD('root') where user='root';

2.flush privileges;

#### 忘记数据库密码怎么办？

第一步：先停掉数据库

[root@db02 ~]# /etc/init.d/mysqld stop

Shutting down MySQL.. SUCCESS!

第二步：无密码、无网络登录

[root@db02 ~]# mysqld\_safe --skip-grant-tables --skip-networking

# mysqld\_safe ==启动数据库服务

#--skip-grant-tables ==mysql的启动一个认证模块==跳过认证模块

#--skip-networking ==关掉TCP/IP,只能本地连接---安全起见

第三步：允许免认证登录

[root@db02 ~]# mysqld\_safe --skip-grant-tables --skip-networking &

[root@db02 ~]# mysql

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 1

Server version: 5.6.38 Source distribution

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

第四步： 进入MySQL系统

1 #无密码登录

mysql

2）#进入到删除用户的数据库，后续也可以用绝对路径==mysql.user

mysql> use mysql

第五步： 修改MySQL密码

mysql> update user set password=PASSWORD(‘123’) where user=’root’ and host=’localhost’;

#### 误删除所有用户的操作流程

第一步

/etc/init.d/mysqld stop

第二步：

mysqld\_safe --skip-grant-tables --skip-networking &

第三步

1 #无密码登录

mysql

2）#进入到删除用户的数据库，后续也可以用绝对路径==mysql.user

mysql> use mysql

3)#user表中插入原先的值

mysql> insert into user(user,host,password,ssl\_type,ssl\_cipher,x509\_issuer,x509\_subject) values(‘root’,’localhost’,PASSWORD(‘123’),’’,’’,’’,’’);

4）/etc/init.d/mysqld restart

5)查看并使用原用户和密码登录确认

#查看用户是否已经创建

mysql> select \* from mysql.user \G;

#用原先的用户及密码登录

#### 误删除所有用户或者忘记root密码

1. 第一步

/etc/init.d/mysqld stop

第二步：启动特殊模式

mysqld\_safe --skip-grant-tables --skip-networking &

第三步

1 #无密码登录

mysql

2）#进入到删除用户的数据库，后续也可以用绝对路径==mysql.user

mysql> use mysql

3）删除原先有问题的数据

mysql> delete from user;

4)向数据库中插入用户名、密码等信息数据库

mysql> INSERT INTO mysql.user VALUES('localhost',

-> 'root',PASSWORD('123'),'Y','Y','Y','Y',

-> 'Y','Y','Y','Y','Y','Y','Y','Y','Y',

-> 'Y','Y','Y','Y','Y','Y','Y','Y','Y',

-> 'Y','Y','Y','Y','Y','Y','Y','','','','',

-> 0,0,0,0,'mysql\_native\_password','','N');

Query OK, 1 row affected (0.00 sec)

mysql> select \* from mysql.user \G;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Host: localhost

User: root

Password: \*23AE809DDACAF96AF0FD78ED04B6A265E05AA257

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

Super\_priv: Y

Create\_tmp\_table\_priv: Y

Lock\_tables\_priv: Y

Execute\_priv: Y

Repl\_slave\_priv: Y

Repl\_client\_priv: Y

Create\_view\_priv: Y

Show\_view\_priv: Y

Create\_routine\_priv: Y

Alter\_routine\_priv: Y

Create\_user\_priv: Y

Event\_priv: Y

Trigger\_priv: Y

Create\_tablespace\_priv: Y

ssl\_type:

ssl\_cipher:

x509\_issuer:

x509\_subject:

max\_questions: 0

max\_updates: 0

max\_connections: 0

max\_user\_connections: 0

plugin: mysql\_native\_password

authentication\_string:

password\_expired: N

1 row in set (0.00 sec)

5)重启数据库

[root@db02 ~]# /etc/init.d/mysqld restart

6)登入数据库验证是否更改完成

mysql> mysql -uroot -p123

## 权限基本管理（详情参考“基础操作==用户权限管理”）

### mysql权限定义

PS：只有本地的root用户才有授权的权限

1）作用对象

库、表

2）权限

INSERT，SELECT，UPDATE

3）归属

每次设定只能有一个“属主”，没有属组或其他人概念

### 权限设定方法

#### #掌握一条命令即可

grant select,update,delete,insert on oldboy.\* to [oldboy@’10.0.0.%’identified](mailto:oldboy@'10.0.0.%25'identified) by ‘123’;

#### #权限命令说明：

mysql> grant all on wordpress.\* to [wordpress@’10.0.0.5%](mailto:wordpress@'10.0.0.5%)’ identified by ‘abc’

命令说明：

1）#all==权限

2）# wordpress.\*==作用对象

#作用对象分解

---\*.\*

------全局级别：当前mysql中所有的库下的所有表，范围是全局的

--wordpress.\*

-----单库级别：wordpress单库下的所有表范围

--wordpress.t1

----单表级别：只对t1表设定权限

3）[#wordpress@’10.0.0.5%](mailto:#wordpress@'10.0.0.5%)’==归属

4）#abc==密码

### 权限设定及演示举例

#### 创建模拟环境

环境：

------------------------------------------------

wordpress -----表t1 和 表t2

blog ------表tb1

------------------------------------------------

操作指令：

mysql> create database wordpress

mysql> use wordpress;

mysql> create table t1 (id int);

mysql> create table t2 (id int);

mysql> create database blog;

mysql> use blog;

mysql> create table tb1(id int);

#### 操作授权命令

#全局级别，wordpress用户对所有库下的所有表均有读的权限

grant select on \*.\* to wordpress@'10.0.0.5%' identified by '123';

#单库级别：wordpress用户对wordpress库下的所有表均有插入、删除、更新权限

grant insert,delete,update on wordpress.\* to wordpress@'10.0.0.5%' identified by '123';

#单表级别，wordpress用户对wordpress库下的t1表拥有所有权限

grant all on wordpress.t1 to wordpress@'10.0.0.5%' identified by '123';

#### 假设

一个客户端程序使用wordpress用户从10.0.0.59登录到mysql中后

Q1：对t1表的管理能力

Q2：对t2表的管理能力

Q3：对tb1表的管理能力

A1：所有权限

A2：select、insert、delete、update

A3：select

#### 测试验证

#### 结论

在不同界别都包含某个表的管理能力时，权限是相加关系，但是我们不

推荐在多级别定义重复权限，

最常用的权限设定方式是对单库进行授权，即：wordpress.\*

### 权限设定后如何回收

revoke select on wordpress.\* from root@’10.0.0.%’;

## 连接管理

### 自带工具连接-mysql

[root@db02 bin]# ll /application/mysql/bin/mysql

-rwxr-xr-x 1 mysql mysql 9822705 Oct 2 17:36 /application/mysql/bin/mysql

[root@db02 bin]# file /application/mysql/bin/mysql

/application/mysql/bin/mysql: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.18, not stripped

# LSB executable==二进制可执行文件

### 自带工具连接-mysql的使用

#### @两种常见的选项语法格式

--长格式----shell> mysql --<option>

--简易格式----shell> mysql -<option>

#### @mysql命令的使用

1.从命令行调用

--linux或unix shell提示符（终端窗口）

----shell> mysql [options]

2.查看帮助信息

----shell> mysql –help（或-？）

3.查看客户端程序的版本

----shell> mysql –version （-V）

### 连接 args or option常见特定于客户机的连接

-u ==--user==user

-p ==password

-h ==mysql的IP地址

--protocol

-P ==port==端口号

-S ==socket文件

-e ==免交互执行mysql命令

### 连接 CMD

#只限定在没有设定任何密码时可以本地登录的方式

mysql

#应用远程：tcp/ip连接串模式=通用的链接模式

mysql -uroot -poldboy123 -h 10.0.0.51 -p3306 [–protocol TCP]

或者mysql -uroot -poldboyw123 -h 10.0.0.51 -P 3306

#本地socket连接到同一主机上运行的服务器

mysql -uroot -poldboy123 -S /application/mysql/tmp/mysql.sock

#远程连接到在其他主机上运行的服务器

mysql -uroot -poldboy123 -e “select user,host,password from mysql.user”

## mysqladmin

### mysqladmin version

root@db01 ~]# mysqladmin -uroot -p version

Enter password:

mysqladmin Ver 8.42 Distrib 5.6.38, for Linux on x86\_64

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Server version 5.6.38

Protocol version 10

Connection Localhost via UNIX socket

UNIX socket /application/mysql-5.6.38/tmp/mysql.sock

Uptime: 26 sec

Threads: 1 Questions: 2 Slow queries: 0 Opens: 70 Flush tables: 1 Open tables: 63 Queries per second avg: 0.076

### mysqladmin status

[root@db01 ~]# mysqladmin -uroot -p status

Enter password:

Uptime: 51 Threads: 1 Questions: 4 Slow queries: 0 Opens: 70 Flush tables: 1 Open tables: 63 Queries per second avg: 0.078

### mysqladmin ping

[root@db01 ~]# mysqladmin -uroot -p ping

Enter password:

mysqld is alive

### mysqladmin processlist

[root@db01 ~]# mysqladmin -uroot -p processlist

Enter password:

+----+------+-----------+----+---------+------+-------+------------------+

| Id | User | Host | db | Command | Time | State | Info |

+----+------+-----------+----+---------+------+-------+------------------+

| 5 | root | localhost | | Query | 0 | init | show processlist |

+----+------+-----------+----+---------+------+-------+------------------+

[root@db01 ~]#

### mysqladmin shutdown

[root@db01 ~]# mysqladmin -uroot -p shutdown

Enter password:

[root@db01 ~]# ps -ef|grep mysql

root 2939 2918 0 06:54 pts/0 00:00:00 mysql -uroot -p

root 3228 2940 0 06:58 pts/1 00:00:00 grep --color=auto mysql

[root@db01 ~]#

### mysqladmin variables

[root@db01 ~]# mysqladmin -uroot -p variables

# 未完待续

## socket文件在编译安装时，已经自定义了位置

#查看二进制文件mysql中是否已经预先自定制了mysql.sock文件位置

[root@db02 bin]# strings /application/mysql/bin/mysql|grep sock

socket

getsockopt

setsockopt

getsockname

vio\_socket\_timeout

socket lH

socket lH

UNIX socket: %s

The protocol to use for connection (tcp, socket, pipe, memory).

The socket file to use for connection.

The buffer size for TCP/IP and socket communication.

/application/mysql-5.6.38/tmp/mysql.sock

Can't create UNIX socket (%d)

Localhost via UNIX socket

Can't connect to local MySQL server through socket '%-.100s' (%d)

Can't create TCP/IP socket (%d)

/server/tools/mysql-5.6.38/vio/viosocket.c

#那么问题来了：如何在mysql安装好了之后，再次修改mysql.sock的位置？？

# 参考资料

[1]https://blog.csdn.net/u010342038/article/details/52798320?utm\_source=copy

[2]https://blog.csdn.net/sangjinchao/article/details/60324724?utm\_source=copy