

4. MySQL基础管理✓

5.1 Mysql用户的介绍

5.1.1 用户的功能

- 1.登陆数据库
- 2.管理数据库对象

5.1.2 用户的组成

用户名@'白名单'

什么是白名单呢？

- 白名单写的是地址列表
支持的格式如下
- 1.% 所有网络用户可以登陆，不包含本地用户
 - 2.10.0.0.10 单一地址
 - 3.10.0.0.% 一个网段
 - 4.10.0.0.5% 10个地址
 - 5.10.0.0.0/255.255.254.0(23位掩码)
 - 6.dba.com 域名
 - 7.db01 主机名
 - ###本地连接
 - 8.127.0.0.1
 - 9.localhost ==socket

5.2 Mysql用户的管理

5.2.1 创建用户

先建立用户，再给用户授权

- 创建用户三种方式+特殊方式
- 1.创建无密码用户

```
mysql> create user cry@'10.0.0.%';
```
 - 2.创建有密码用户(identified by)

```
mysql> create user cry@'10.0.0.%' identified by '123';
```
 - 3.创建有密码用户时指定加密插件(with)

```
mysql> create user cry@'10.0.0.%' identified with mysql_native_password by '123';
```
 - 4.mysql8.0版本之前 授权和创建用户操作可同时 (grant)

```
mysql> grant all on *.* to cry@'10.0.0.%' identified by '123';
```

8.0版本不支持grant 命令后使用 identified by语句

5.2.2 查询用户

DBA日常工作巡检中要检查“无头无尾的用户”，进行修改删除
root@'localhost' 创建mysql默认的超级管理员

Bash | Copy

```
1 mysql> select user,host,authentication_string,plugin from mysql.user;
2
3 | user | host | authentication_string | plugin
4
5 | oldguo | 10.0.0.% | *23AE809DDACAF96AF0FD78ED04B6A265E05AA257 | mysql_native_password
6 | mysql.infoschema | localhost | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | cacti
7 | mysql.session | localhost | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | cacti
8 | mysql.sys | localhost | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | cacti
9 | root | localhost | | cacti
10
11 5 rows in set (0.00 sec)
```

5.2.3 删除用户 生产环境中不要轻易使用!!! 我们可以进行锁定用户

删除用户命令 (删除用户会同时删除用户的权限)

```
drop user oldguo@'10.0.0.%';
用户锁定和解锁命令
mysql> alter user oldguo@'10.0.0.%' account lock;
mysql> alter user oldguo@'10.0.0.%' account lock;
```

5.2.4 修改用户

```
#可以修改用户各个信息
#修改root用户密码
mysql> alter user root@'localhost' identified by '123';
```

5.2.5 用户资源管理

密码过期使用

```
查看密码的生命周期
mysql> select @@default_password_lifetime;
+-----+
| @@default_password_lifetime |
+-----+
| 0 |
```

```
+-----+
修改密码的生命周期
mysql> SET PERSIST default_password_lifetime = 180;
永不限期
CREATE USER 'oldguo'@'localhost' PASSWORD EXPIRE NEVER;
ALTER USER 'oldguo'@'localhost' PASSWORD EXPIRE NEVER;
```

密码重用（不常用）

```
1.提示最近几次密码不能再使用
password_history=6 最近6次密码不能再使用
2.最近密码不能重用的天数
password_resuse_interval=365 (天)
```

锁定用户

```
mysql> alter user oldguo@'10.0.0.0%' account lock;
mysql> alter user oldguo@'10.0.0.0%' account lock;
```

连接资源限制（不常用）

```
with后跟上
MAX_QUERIES_PER_HOUR count 规定一小时查询的次数
MAX_UPDATES_PER_HOUR count
每个小时连接次数的综合
MAX_CONNECTIONS_PER_HOUR 2000 针对所有用户
MAX_USER_CONNECTIONS count 针对单个用户
```

5.2.6 用户管理规范

```
1.专用户专业务
2.密码复杂程度
3.删除用户使用锁定用户方式替代(巡检发现“无头无尾用户”)
4.细化用户的白名单
```

5.3 Mysql权限的介绍

5.3.1 权限的对象

```
8.0版本之前：授权命令基础权限
8.0版本之后：出现role角色概念（常用命令基础权限的打包），可以授权给角色
```

5.3.2 权限级别

```
1.库级别：
• 全库级别 *.*
```

- 单库级别 db.*

2.表级别:

- db.t1

3.列级别:

select (id,name)

5.4 Mysql权限的管理

5.4.1 授权表

show privileges; 查看当前数据库可以授权的权限选项

Bash | Copy

```

1  mysql> show privileges;
2  +-----+-----+-----+
3  | Privilege           | Context                | Comment
4  +-----+-----+-----+
5  | Alter               | Tables                 | To alter the table
6  | Alter routine       | Functions,Procedures  | To alter or drop stored functions/p
7  | Create              | Databases,Tables,Indexes | To create new databases and tables
8  | Create routine      | Databases              | To use CREATE FUNCTION/PROCEDURE
9  | Create role         | Server Admin           | To create new roles
10 | Create temporary tables | Databases              | To use CREATE TEMPORARY TABLE
11 | Create view         | Tables                 | To create new views
12 | Create user         | Server Admin           | To create new users
13 | Delete              | Tables                 | To delete existing rows
14 | Drop                | Databases,Tables       | To drop databases, tables, and view
15 | Drop role           | Server Admin           | To drop roles
16 | Event               | Server Admin           | To create, alter, drop and execute
17 | Execute             | Functions,Procedures  | To execute stored routines
18 | File                | File access on server  | To read and write files on the serv
19 | Grant option        | Databases,Tables,Functions,Procedures | To give to other users those privil
20 | Index               | Tables                 | To create or drop indexes
21 | Insert              | Tables                 | To insert data into tables
22 | Lock tables         | Databases              | To use LOCK TABLES (together with S
23 | Process             | Server Admin           | To view the plain text of currently
24 | Proxy               | Server Admin           | To make proxy user possible
25 | References          | Databases,Tables       | To have references on tables
26 | Reload              | Server Admin           | To reload or refresh tables, logs a
27 | Replication client  | Server Admin           | To ask where the slave or master se
28 | Replication slave   | Server Admin           | To read binary log events from the
29 | Select              | Tables                 | To retrieve rows from table
30 | Show databases      | Server Admin           | To see all databases with SHOW DATA
31 | Show view           | Tables                 | To see views with SHOW CREATE VIEW
32 | Shutdown            | Server Admin           | To shut down the server
33 | Super               | Server Admin           | To use KILL thread, SET GLOBAL, CHA
34 | Trigger             | Tables                 | To use triggers
35 | Create tablespace   | Server Admin           | To create/alter/drop tablespaces
36 | Update              | Tables                 | To update existing rows
37 | Usage               | Server Admin           | No privileges - allow connect only
38 | XA_RECOVER_ADMIN    | Server Admin           |
39 | SHOW_ROUTINE        | Server Admin           |
40 | RESOURCE_GROUP_USER | Server Admin           |
41 | SET_USER_ID         | Server Admin           |
42 | SESSION_VARIABLES_ADMIN | Server Admin           |
43 | CLONE_ADMIN         | Server Admin           |
44 | PERSIST_RO_VARIABLES_ADMIN | Server Admin           |
45 | ROLE_ADMIN          | Server Admin           |
46 | BACKUP_ADMIN        | Server Admin           |
47 | CONNECTION_ADMIN    | Server Admin           |
48 | RESOURCE_GROUP_ADMIN | Server Admin           |
49 | INNODB_REDO_LOG_ARCHIVE | Server Admin           |
50 | BINLOG_ENCRYPTION_ADMIN | Server Admin           |

```

51	GROUP_REPLICATION_ADMIN	Server Admin
52	SYSTEM_USER	Server Admin
53	APPLICATION_PASSWORD_ADMIN	Server Admin
54	TABLE_ENCRYPTION_ADMIN	Server Admin
55	SERVICE_CONNECTION_ADMIN	Server Admin
56	AUDIT_ADMIN	Server Admin
57	BINLOG_ADMIN	Server Admin
58	ENCRYPTION_KEY_ADMIN	Server Admin
59	SYSTEM_VARIABLES_ADMIN	Server Admin
60	REPLICATION_SLAVE_ADMIN	Server Admin

5.4.2 授权操作

授于权限

授权的语法格式

```
mysql> grant + all或者应用权限或单独某个列+ on +权限级别+to+用户名@'白名单';
```

#授权all

```
mysql> grant all on *.* to oldboy@'10.0.0.%' ;
```

#授权应用权限（增删改查）

```
mysql> grant select ,update ,delete ,insert on *.* to oldguo@'10.0.0.%' ;
```

#授权单独某个列

```
mysql> grant select(id) on *.* to oldguo@'10.0.0.%';
```

角色授权8.0新特性!!!

1.创建角色

```
mysql> create role dev@'10.0.0.%';
```

2.对创建的角色进行授权

```
mysql> grant select on *.* to dev@'10.0.0.%';
```

3.再把授权后角色授权给用户

```
mysql> grant dev to user2@'10.0.0.%';
```

4.查看相关角色权限命令

```
mysql.role_edges;
```

```
information_schema.user_privileges;
```

查看权限

语法格式

```
mysql> show grants for 用户名@'白名单'
```

#举例

```
mysql> show grants for oldguo@'10.0.0.%';
```

```
+-----+
| Grants for oldguo@10.0.0.% |
```

```
+-----+
| GRANT USAGE ON *.* TO `oldguo`@`10.0.0.%` |
+-----+
```

回收权限

语法格式
revoke +应用权限+ on+权限级别+用户名@'白名单'
#举例
mysql> revoke delete on *.* from oldguo@'10.0.0.%';

5.4.3 权限管理规范

不同用户之间授权的权限区别
管理员用户: all (去掉drop)
开发用户: Create ,Create routine,Create temporary tables,Create view,Delete ,Event ,Execute,Insert ,References,Select,Show databases ,Show view ,Trigger,Update
监控用户: select , replication slave , client supper
备份用户: all
主从用户: replication slave
业务用户: insert , update , delete ,select

5.5 Mysql连接管理

本地连接

5.5.1 socket

前提:必须创建和授权localhost相关用户

```
#创建oldguo本地连接用户
mysql> create user oldguo@'localhost' identified by '123';
#授予oldguo本地连接用户权限
mysql> grant all on *.* to oldguo@'localhost' with grant option;
```

本地连接用户方式

```
[root@db01 ~]# mysql -u oldguo -p123 -S /tmp/mysql.sock
```

远程连接

5.5.2 网络连接 Tcp/Ip

前提: 必须创建和授权远程网段相关用户

```
#创建oldguo远程连接相关用户
```

```
mysql> create user oldguo@'10.0.0.%' identified by '123';  
#给oldguo创建的远程连接用户进行授权  
mysql> grant all on *.* to oldguo@'10.0.0.%' with grant option; （一般业务用户使用这种方式，授权给与增删改查权限就可）
```

远程连接用户方式

```
[root@db01 ~]# mysql -uoldguo -p123 -h 10.0.0.51 -P3306
```

5.5.3 客户端工具连接

- 1.sqlyong [sqlyog10.zip \(20.5 MB\)](#) （第三方）
- 2.navicat （第三方）
- 3.workbench(官方工具)

5.5.4 基于ssl的安全连接（数据库暴漏在外网环境下）

查看mysql中关于ssl功能的参数，8.0版本默认开启

```
mysql> show variables like '%ssl%';
```

Variable_name	Value
have_openssl	YES
have_ssl	YES
mysqlx_ssl_ca	
mysqlx_ssl_capath	
mysqlx_ssl_cert	
mysqlx_ssl_cipher	
mysqlx_ssl_crl	
mysqlx_ssl_crlpath	
mysqlx_ssl_key	
ssl_ca	ca.pem
ssl_capath	
ssl_cert	server-cert.pem
ssl_cipher	
ssl_crl	
ssl_crlpath	
ssl_fips_mode	OFF
ssl_key	server-key.pem

使用ssl连接数据库

```
mysql -uroot -p123 -h10.0.0.51  
--ssl-cert=/data/mysql/data_3306/client-cert.pem --ssl-key=/data/mysql/data_3306/client-key.pem
```

指定数据目录下的公钥

指定数据目录下的私钥

5.6 Mysql8.0初始化配置文件管理

5.6.1 配置文件的作用

规范：数据库先规划配置文件，在初始化

当数据库启动时，会自动加载配置文件，从而配置文件中的参数影响数据库。也会影响本地连接功能

5.6.2 配置文件的构成

标签项+配置项两部分组成

```
[标签]
配置=xxx
配置=xxx
配置=xxx
[标签]
.....
```

标签项

- 作用：分类不同程序下的参数

服务端标签

- 【server】 所有服务器端程序运行时会调用标签下的所有参数配置 （不建议使用）
- 【mysqld】 影响数据库启动， 和数据库初始化 （常用）
- 【mysqld_safe】 影响数据库启动，比mysqld程序中参数更加丰富 （常用）

客户端标签（不影响远程登陆，只影响本地登陆）

- 【client】 包含所有客户端参数 （不建议使用）
- 【mysql】 本地登陆数据库时会使用这个程序
- 【mysqldump】 本地备份时会使用这个程序

配置项

```
配置模板构成
[mysqld]                ##服务端
user=mysql              # 管理用户
basedir=/usr/local/mysql  # 软件程序路径位置
datadir=/data/3306/data  # 数据存放路径位置
socket=/tmp/mysql.sock   # 指定套接字文件socket在那个路径下生成
server_id=6              #服务器ID,主从时标识不同主机
log_bin=/data/mysql/binlog_3306 #二进制日志存放位置路径
port=3306                #端口
[mysql]                 ##客户端
socket=/tmp/mysql.sock   #本地登陆指定使用socket文件的路径
```

如果我们先直接使用mysql本地登陆，可以在配置文件【mysql】标签下写关于登陆数据的用户信息和密码信息即可。

```
[mysql]
socket=/tmp/mysql.sock
user=root
password=123
```

```
[root@db01 ~]# cat /etc/my.cnf
[mysqld]
user=mysql
basedir=/usr/local/mysql
datadir=/data/3306/data
socket=/tmp/mysql.sock
[mysql]
socket=/tmp/mysql.sock
user=root
password=123
[root@db01 ~]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.20 MySQL Community Server - GPL

Copyright (c) 2000, 2020, 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>
```

5.6.3 配置文件路径的选择

数据库在读取配置文件路径时会有优先级

```
1 my.cnf可以在四个路径下数据库进行读取
2
3 [root@db01 ~]# mysqld --help --verbose | grep my.cnf
4 /etc/my.cnf /etc/mysql/my.cnf /usr/local/mysql/etc/my.cnf ~/.my.cnf
   优先级从左至右依次读取，当发生冲突会发生覆盖，以最后读取的文件为准
```

手动指定默认路径配置文件的方法（会在多实例中应用）

```
mysql(mysql_safe) --defaults-file=指定的配置文件路径 &
```

5.7 Mysql多种启动关闭方式

启动

Mysql的多种启动方式，归根揭底是调用mysql这个核心守护进程

启动方式一：

mysql是mysql核心守护进程，路径在 /usr/local/mysql/bin/mysql

```
]# mysql & : 启动方式会把所有启动日志全部打印在屏幕上，不便管理员做日志收集
```

启动方式二：

```
]# mysql_safe & : 会把所有启动日志记录在数据目录下的db01.err错误日志中,将启动结果打印在屏幕上
```

补充：mysql_safe &和mysql & 关系

优点：可以启动时添加参数

缺点：没有提供关闭的方法

mysql_safe 会监控mysql，当发现mysql死亡，调用mysql起来

启动方式三：

mysql自己提供的启动脚本，存放在 /usr/local/mysql/support-files 下的 mysql.server

我们就可以相对路径使用这个脚本对mysql进行启动关闭管理了

```
[root@db01 support-files]# ./mysql.server start(restart,stop)
```

启动方式四：

我们可以把这个mysql.server的脚本移动到linux操作系统专门存放服务启动脚本的路径下(/etc/init.d/mysql)，方便我们管理 cp /usr/local/

mysql/support-files/mysql.server /etc/init.d/mysql

在/etc/init.d/mysql这个路径下调用即可 /etc/init.d/mysql start(restart,stop)

启动方式五：

把脚本拷贝到/etc/init.d/mysql也是为了迎合操作系统管理服务的这种管理模式。

操作系统管理模式分为两种

centos6管理模式是 service+服务+start

centos6模式下mysql启动方式是 service mysql start (restart,status,stop)

centos7管理模式是systemctl+服务+start

如果想使用，先将mysql加入下次开机自启 systemctl enable mysql

```
[root@db01 init.d]# systemctl enable mysql
mysql.service is not a native service, redirecting to /sbin/chkconfig.
Executing /sbin/chkconfig mysql on
```

就可以使用systemctl这种模式了 `systemctl mysqld start(restart ,status,stop)`

总结多种启动方式关系!!!

多种启动方式的逻辑关系，是层层调用最终调用核心守护进程mysqld

systemctl(service)--调用-->/etc/init.d/mysqld--调用-->mysql.server(脚本)--调用-->mysqld_safe & --监控调用--> mysqld &

关闭

关闭方式一：

标准关闭： `]# mysqladmin -uroot -p 123 shutdown` (内部原理也是用到kill的方式，不过kill级别比较低)

关闭方式二：

登陆到数据库中，进行 `mysql>shutdown;` 操作,8.0版本支持登陆进去 `mysql>restart;` 操作。

00908aaca299.png&title=4.%20MySQL%E5%9F%BA%E7%A1%80%E7%AE%A1%E7%90%86%E2%88%9A%20%7C%205.1%20Mysql%E7%94%A8%E6%88%B7%E7%9A%84%E4%BB%8B%E7%BB%8D5.1.1