# 目录

# MySQL 数据库安全

# 目录

# MySQL 基本操作

# 为初始安装的mysql设置登录密码并登录

使用MySQL自带的命令mysqladmin设置root密码

```
# mysqladmin -u root password "123456.."
```

使用mysqladmin命令更改root密码

```
# mysqladmin -u root -p123456.. password "1234567.."
```

旧密码　　新密码

登录MySQL

```
#mysql -u root -p
```

```
[root@localhost 桌面]# mysqladmin -u root password "123456.."
[root@localhost 桌面]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.1.71 Source distribution

Copyright (c) 2000, 2013, 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>
```

# 数据库管理

| MySQL命令 | 功能 |
| --- | --- |
| show databases; | 查看服务器中当前有哪些数据库 |
| use 数据库名; | 选择所使用的数据库 |
| create database 数据库名; | 创建数据库 |
| drop database 数据库名; | 删除指定的数据库 |

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| test               |
+--------------------+
3 rows in set (0.00 sec)

mysql>
```

mysql安装后默认会创建三个数据库information_schema、mysql和test, 其中名为“mysql”的数据库很重要,它里面保存有MYSQL的系统信息,用户修改密码和新增用户,实际上就是针对该数据库中的有关数据表进行操作。

# 数据表结构管理

| MySQL命令 | 功能 |
| --- | --- |
| create table 表名 (字段设定列表); | 在当前数据库中创建数据表 |
| show tables; | 显示当前数据库中有哪些数据表 |
| describe [数据库名.]表名; | 显示当前或指定数据库中指定数据表的结构(字段)信息 |
| drop table [数据库名.]表名; | drop table [数据库名.]表名; |

创建表的时候必须至少有一个列

```
mysql> create table wt;
ERROR 1113 (42000): A table must have at least 1 column
mysql> create table wt(id char(10));
Query OK, 0 rows affected (0.06 sec)

mysql> describe wt;
+-------+----------+------+-----+---------+-------+
| Field | Type     | Null | Key | Default | Extra |
+-------+----------+------+-----+---------+-------+
| id    | char(10) | YES  |     | NULL    |       |
+-------+----------+------+-----+---------+-------+
1 row in set (0.00 sec)

mysql>
```

# 记录的查看、插入、修改与删除

| MySQL命令 | 功能 |
| --- | --- |
| insert into 表名(字段1,字段2,……) values(字段1的值, 字段2的值,……); | 向数据表中插入新的记录 |
| update 表名 set 字段名1=字段值1[,字段名2=字段值2] where 条件表达式; | 修改、更新数据表中的记录 |
| select 字段名1,字段名2……from 表名 where 条件表达式; | 从数据表中查找符合条件的记录 |
| select * from 表名; | 显示当前数据库的表中的记录 |
| delete from 表名 where 条件表达式; | 在数据表中删除指定的记录 |
| delete from 表名; | 将当前数据库表中记录清空 |

# mysql基本操作

查看用户及host等信息

mysql>select user,host,password from mysql.user;

```
mysql> select user, host, password from mysql.user;
+------+-----------------------+-------------------------------------------+
| user | host                  | password                                  |
+------+-----------------------+-------------------------------------------+
| root | localhost             | *F7C500C92170650F5A1A5D09F5B80C1C9BEA631C |
| root | localhost.localdomain |                                           |
| root | 127.0.0.1             |                                           |
|      | localhost             |                                           |
|      | localhost.localdomain |                                           |
+------+-----------------------+-------------------------------------------+
5 rows in set (0.00 sec)
```

新建用户

```
mysql> grant all privileges ON *.* TO 'test'@'localhost'
    -> identified by '123456' with grant option;
Query OK, 0 rows affected (0.00 sec)

mysql> select user, host, password from mysql.user;
+------+-----------------------+-------------------------------------------+
| user | host                  | password                                  |
+------+-----------------------+-------------------------------------------+
| root | localhost             | *F7C500C92170650F5A1A5D09F5B80C1C9BEA631C |
| root | localhost.localdomain |                                           |
| root | 127.0.0.1             |                                           |
|      | localhost             |                                           |
|      | localhost.localdomain |                                           |
| test | localhost             | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
+------+-----------------------+-------------------------------------------+
6 rows in set (0.00 sec)
```

# grant命令

```
mysql> grant all privileges ON *.* TO 'test'@'localhost'
    -> identified by '123456' with grant option;
```

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。
test@'localhost' 表示test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。
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他用户授权。
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了一个select权限，然后又给用户添加了一个insert权限，那么该用户就同时拥有了select和insert权限

# mysql基本操作

查询、增加、取消用户权限

mysql>show grants for test@localhost;

```
mysql> show grants for test@localhost;
+-----------------------------------------------------------------------------------------------------
-----------------------------------------+
| Grants for test@localhost
                                         |
+-----------------------------------------------------------------------------------------------------
-----------------------------------------+
| GRANT ALL PRIVILEGES ON *.* TO 'test'@'localhost' IDENTIFIED BY PASSWORD '*6BB4837EB74329105EE4
568DDA7DC67ED2CA2AD9' WITH GRANT OPTION |
+-----------------------------------------------------------------------------------------------------
-----------------------------------------+
1 row in set (0.00 sec)
```

mysql>grant select,insert,delete on *.* to test@localhost;

mysql> revoke delete ON *.* from test@localhost;

```
mysql> show grants for test@localhost;
+-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------+
| Grants for test@localhost
                                                                                                                                                           |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------+
| GRANT SELECT, INSERT, UPDATE, CREATE, DROP, RELOAD, SHUTDOWN, PROCESS, FILE, REFERENCES, INDEX, ALTER, SHOW DATABASES, SUPER, CREATE TEMPORA
RY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EV
ENT, TRIGGER ON *.* TO 'test'@'localhost' IDENTIFIED BY PASSWORD '*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9' WITH GRANT OPTION |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)
```

# 用户权限

MySQL可以为不用的用户分配严格的、复杂的权限、这些操作大多都可以用sql指令grant（分配权限）和revoke(回收权限)来实现。

| | |
|---|---|
| ALTER： | 修改表和索引 |
| CREATE： | 创建数据库和表 |
| DROP: | 抛弃（删除）数据库和表 |
| INDEX： | 创建或抛弃索引 |
| INSERT： | 向表中插入新行 |
| SELECT： | 检索表中的记录 |
| UPDATE | 修改现存表记录 |
| FILE： | 读或写服务器上的文件 |
| PROCESS： | 查看服务器中执行的线程信息或杀死线程 |
| RELOAD： | 重载授权表或清空日志。主机缓存或表缓存 |
| SHUTDOWN： | 关闭服务器 |
| ALL： | 所有权限，ALL PRIVILEGES同义词 |
| USAGE： | 特殊的“无权限”权限 |

# MySQL 安全加固

# 删除默认数据库和数据库用户

MySQL初始化后会自动生成空用户和test库，进行安装的测试，这会对数据库的安全构成威胁，有必要全部删除，最后的状态只保留单个root即可，当然以后根据需要增加用户和数据库。

```
mysql> show databases;
mysql> drop database test;                    //删除数据库test
mysql> use mysql;
mysql> delete from db;                        //删除存放数据库的表信息，因为还没有数据库信息。
mysql> delete from user where not (user='root');          // 删除初始非root的用户
mysql> delete from user where user='root' and password=''; //删除空密码的root尽量重复操作
```

```
mysql> select user, host, password from user ;
+------+-----------+-------------------------------------------+
| user | host      | password                                  |
+------+-----------+-------------------------------------------+
| root | localhost | *F7C500C92170650F5A1A5D09F5B80C1C9BEA631C |
+------+-----------+-------------------------------------------+
```

```
mysql> flush privileges;                      //强制刷新内存授权表。
```

# 不使用默认密码和弱口令

检查账户默认密码和弱口令，口令长度至少8位，并包括数字、小写字母、大写字母和特殊符号四类中至少两种。且5次以内不得设置相同的口令，密码应至少90天更换一次。

Mysql> Update user set password=password(‘test!p3’) where user=‘root’;
Mysql> Flush privileges;

检测操作
检查本地密码： (注意，管理帐号 root 默认是空密码)
mysql> use mysql;
mysql> select Host,User,Password,Select_priv,Grant_priv from user;

# 改变默认mysql管理员帐号

改变默认的mysql管理员账号也可以使mysql数据库的安全性有较好的提高，因为默认的mysql管理员的用户名都是root

mysql> update mysql.user set user='admin' where user='root';

```
mysql> select user, host, password from user ;
+-------+-----------+-------------------------------------------+
| user  | host      | password                                  |
+-------+-----------+-------------------------------------------+
| admin | localhost | *F7C500C92170650F5A1A5D09F5B80C1C9BEA631C |
+-------+-----------+-------------------------------------------+
1 row in set (0.00 sec)
```

# 使用独立用户运行MySQL

绝对不要作为使用root用户运行MySQL服务器。这样做非常危险，因为任何具有FILE权限的用户能够用root创建文件(例如，~root/.bashrc)。mysqld拒绝使用root运行，除非使用–user=root选项明显指定。应该用普通非特权用户运行 mysql。为数据库建立独立的linux中的mysql账户，该账户用来只用于管理和运行MySQL。

要想用其它linux用户启动mysql，增加user选项指定/etc/my.cnf选项文件或服务器数据目录的my.cnf选项文件中的[mysqld]组的用户名。

#vi /etc/my.cnf
[mysqld]
user=mysql

该命令使服务器用指定的用户来启动，无论你手动启动或通过mysqld_safe或mysql.server启动，都能确保使用mysql的身份。

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
~
```

# 禁止远程连接数据库

使用命令netstat 查看默认的3306端口是打开的，此时打开了mysql的网络监听，允许用户远程通过帐号密码连接数本地据库，默认情况是允许远程连接数据的。为了禁止该功能，启动skip-networking，不监听sql的任何TCP/IP的连接，切断远程访问的权利，保证安全性

```
[root@localhost 桌面]# netstat -anpl |grep 3306
tcp        0      0 0.0.0.0:3306                0.0.0.0:*                   LISTEN      3194/mysq
ld
```

# vi /etc/my.cf　　//将#skip-networking注释去掉。

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
skip-networking
[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

# 限制用户连接的数量

数据库的某用户多次远程连接，会导致性能的下降和影响其他用户的操作，有必要对其进行限制。可以通过限制单个账户允许的连接数量来实现，设置my.cnf 文件的mysql中的max_user_connections变量来完成。GRANT语句也可以支持资源控制选项来限制服务器对一个账户允许的使用范围。

#vi /etc/my.cnf
[mysqld]
max_user_connections=2

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
skip-networking
max_user_connections 2
[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
~
```

# 命令历史记录保护

数据库相关的shell操作命令都会分别记录在.bash_history，如果这些文件不慎被读取，会导致数据库密码和数据库结构等信息泄露，而登陆数据库后的操作将记录在.mysql_history文件中，如果使用update表信息来修改数据库用户密码的话，也会被读取密码，因此需要删除这两个文件，同时在进行登陆或备份数据库等与密码相关操作时，应该使用-p参数加入提示输入密码后，隐式输入密码，建议将以上文件置空。

```
# rm .bash_history .mysql_history          //删除历史记录
# ln -s /dev/null .bash_history            //将shell记录文件置空
# ln -s /dev/null .mysql_history           //将mysql记录文件置空
```

# 对重要数据加密存储

MySQL提供了4个函数用于哈希加密：PASSWORD, ENCRYPT, SHA1和MD5。

INSERT INTO table1 (user, password) VALUE ('user1', MD5('password1') )

```
mysql> select md5("admin")
    -> ;
+----------------------------------+
| md5("admin")                     |
+----------------------------------+
| 21232f297a57a5a743894a0e4a801fc3 |
+----------------------------------+
1 row in set (0.09 sec)
```

```
mysql> select password("admin");
+-------------------------------------------+
| password("admin")                         |
+-------------------------------------------+
| *4ACFE3202A5FF5CF467898FC58AAB1D615029441 |
+-------------------------------------------+
1 row in set (0.00 sec)
```

```
mysql> select encrypt("admin");
+------------------+
| encrypt("admin") |
+------------------+
| 3ExeWKRBto4LI    |
+------------------+
1 row in set (0.00 sec)
```

```
mysql> select sha1("admin");
+------------------------------------------+
| sha1("admin")                            |
+------------------------------------------+
| d033e22ae348aeb5660fc2140aec35850c4da997 |
+------------------------------------------+
1 row in set (0.00 sec)

mysql>
```

# 本地文件读取保护

常见攻击：拥有FILE权限

```
CREATE TABLE etc_passwd (pwd_entry TEXT);
LOAD DATA INFILE "/etc/passwd" into TABLE etc_passwd;
SELECT * FROM etc_passwd;
```

```
mysql> SELECT * FROM etc_passwd;
+------------------------------------------------------------------------------
-----+
| pwd_entry
     |
+------------------------------------------------------------------------------
-----+
| root:x:0:0:root:/root:/bin/bash
     |
| bin:x:1:1:bin:/bin:/sbin/nologin
     |
| daemon:x:2:2:daemon:/sbin:/sbin/nologin
     |
| adm:x:3:4:adm:/var/adm:/sbin/nologin
     |
| lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
     |
| sync:x:5:0:sync:/sbin:/bin/sync
     |
| shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
     |
| halt:x:7:0:halt:/sbin:/sbin/halt
     |
| mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
     |
| news:x:9:13:news:/etc/news:
     |
| uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
     |
| operator:x:11:0:operator:/root:/sbin/nologin
     |
| games:x:12:100:games:/usr/games:/sbin/nologin
     |
| gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
     |
| ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
     |
| nobody:x:99:99:Nobody:/:/sbin/nologin
     |
```

# 本地文件读取保护

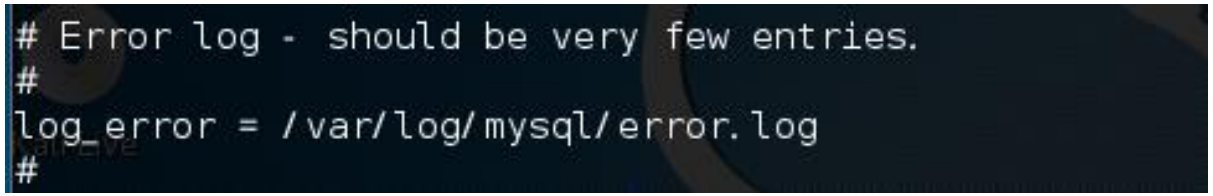
为防止用户读取服务器上的本地文件
防止MySQL使用“LOAD DATA LOCAL INFILE”读取主机上的文件
vim /etc/my.cnf
set-variable=local-infile=0

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
skip-networking
max_user_connections 2
set-variable=local-infile=0
[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

# 日志功能

mysql 有以下几种日志：
错误日志： -log-err
查询日志： -log （可选）
慢查询日志: -log-slow-queries （可选）
更新日志: -log-update
二进制日志： -log-bin
在 mysql 的安装目录下，打开 my.ini,在后面加上上面的参数，保存后重启 mysql 服务就行了。

检查操作
查看etc/my.cnf文件，查看是否包含如下配置：
[mysqld]
Log=filename



# 数据库的备份

1）．直接备份数据库所在的目录
使用cp、tar等命令直接备份数据库所存放的目录

2．使用mysqldump命令备份和恢复
mysqldump -u 用户名 -p [密码] [选项] [数据库名] [表名] > /备份路径/备份文件名

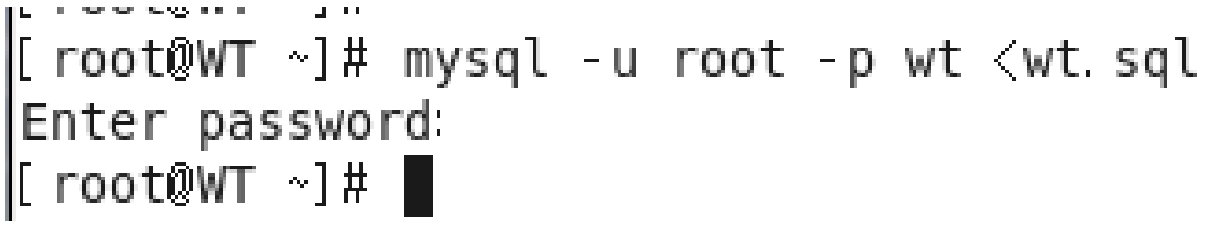
```
mysql> use mysql;
Database changed
mysql>
mysql> show global variables like "%datadir%";
+---------------+-----------------+
| Variable_name | Value           |
+---------------+-----------------+
| datadir       | /var/lib/mysql/ |
+---------------+-----------------+
1 row in set (0.00 sec)
```

```
[root@WT ~]# mysqldump -u root -p wt >wt.sql
Enter password:
[root@WT ~]# cat wt.sql
-- MySQL dump 10.13  Distrib 5.1.71, for redhat-linux-gnu (x86_64)
--
-- Host: localhost    Database: wt
-- ------------------------------------------------------
-- Server version       5.1.71

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0
*/;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

# 数据库的恢复

mysql -u root -p [数据库名] < /备份路径/备份文件名



# THANK YOU